Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 8, No.1, August : 2025

AI-POWERED BLOG PLATFORM WITH REAL-TIME CONTENT MODERATION

Mrs. M. Moohambikai, Assistant Professor, Dept.Of Computer Science, SSM Institute of Engineering and Technology, Dindigul, India. ssmcsemoohambikai@gmail.com
Anand Charukesan K, Computer Science and Engineering, SSM Institute of Engineering and Technology Dindigul, India. anandcharukesan@gmail.com
Indhiraraj S, Computer Science and Engineering, SSM Institute of Engineering and Technology Dindigul, India. indhiraraj7@gmail.com

ABSTRACT
This project presents the design and development of a blog website integrated with AI-powered content moderation to ensure safe and respectful online discourse. The platform allows users to create, edit, and interact with blog content, while employing a hybrid moderation system combining lexicon-based filtering, BERT-based classification, and contextual analysis. These techniques work together to detect explicit, harmful, or offensive language, even in veiled or sarcastic forms. The system is built using Next.js, TypeScript, Tailwind CSS, and MySQL, with Clerk handling authentication and Groq supporting AI inference. The goal is to demonstrate how artificial intelligence can be used to maintain the integrity and quality of user-generated content in modern web applications.

Keywords:
AI moderation, content filtering, BERT, sarcasm detection, Next.js, TypeScript, MySQL, Clerk, Groq, user-generated content.

I. Introduction

In the digital age, user-generated content has become a cornerstone of online communication, with blog platforms offering a dynamic medium for self-expression, knowledge sharing, and community engagement. However, this open format also presents challenges in moderating harmful, offensive, or inappropriate content, which can negatively impact users and the overall platform integrity. To address this issue, this project proposes the development of a blog website integrated with AI-based content moderation capabilities.

Traditional content moderation relies heavily on human oversight, which is time-consuming, inconsistent, and difficult to scale. As platforms grow, manual moderation becomes insufficient, leading to delayed responses and lapses in enforcing community standards. To address these limitations, there is a growing need for intelligent, automated systems capable of detecting and managing harmful content efficiently and in real-time.

The platform enables users to create, edit, and manage blog posts while ensuring the content adheres to community guidelines through automated moderation. The moderation system employs a hybrid approach that combines lexicon-based filtering, BERT-based classification, and contextual analysis to detect explicit, harmful, or veiled offensive language. The goal is to improve content safety without compromising user experience. Developed using Next.js, TypeScript, Tailwind CSS, and MySQL, the system integrates Clerk for secure authentication and Groq for efficient AI inference. This project demonstrates the practical application of artificial intelligence in ensuring content quality and user safety in modern web applications.

An important feature of the proposed platform is its responsiveness across devices and screen sizes. Built with Tailwind CSS and React-based components via Next.js, the user interface adapts seamlessly to mobile phones, tablets, and desktops, ensuring consistent user experience. This responsiveness is essential for modern users who frequently access blogs on-the-go and expect an intuitive, smooth interaction regardless of the device used.

This project proposes the development of a blog website integrated with AI-powered content moderation. The system employs a layered approach that combines lexicon-based filtering, BERT-

based classification, and contextual analysis. Lexicon-based filtering allows quick identification of explicitly harmful keywords. BERT, a transformer-based language model, is used for understanding the context and semantics of user submissions. Additionally, contextual analysis enables the system to interpret veiled threats, sarcasm, and indirect offensive language—cases where traditional filters often fail.

By integrating advanced AI with a user-centric blogging interface, this project aims to demonstrate a practical solution for real-time content moderation that can be adopted in various digital platforms to promote safe and constructive online communities.

II. Literature

Artificial Intelligence (AI) has emerged as a transformative force in the domain of content moderation, especially within user-driven platforms such as blogs, forums, and social media. As digital content becomes more diverse and voluminous, manual moderation is no longer viable. AI-powered systems are now essential to ensure safe, respectful, and policy-compliant online interactions.

Content moderation using AI encompasses a range of technologies, including natural language processing (NLP), machine learning, and deep learning. These systems can analyze large volumes of user-generated content in real time, detecting hate speech, harassment, misinformation, and other forms of harmful communication. Models like BERT (Bidirectional Encoder Representations from Transformers) have revolutionized contextual understanding, allowing platforms to recognize not just explicit language but also sarcasm, metaphor, and implicit abuse.

Moderation is no longer just a technical feature; it is a foundational aspect of maintaining community standards and user trust. Platforms across industries—media, education, e-commerce, and social networks—employ AI moderation to reduce liability, improve user experience, and adhere to regional and international content regulations.

Despite its advantages, AI moderation poses challenges such as bias in datasets, false positives/negatives, and limitations in understanding cultural nuances. To address these, hybrid approaches are increasingly adopted—combining lexicon-based filtering for speed, transformer models for deep understanding, and contextual analysis engines like LLaMA for detecting subtle intent.

In the context of web applications, AI moderation must also be tightly integrated with authentication systems, real-time feedback loops, and scalable backend services. The convergence of frontend responsiveness, backend performance, and intelligent moderation is key to delivering a seamless and secure blogging environment.

As content creation becomes more democratized, the role of AI in ensuring content quality and user safety is no longer optional. Modern blog platforms must leverage intelligent moderation engines not only to prevent harm but also to foster inclusive and constructive digital communities.

2.1 Existing Solutions and Their Limitations

Traditional blog platforms primarily focus on providing users with basic functionalities for creating, reading, updating, and deleting blog posts. However, these systems often lack advanced features to address modern challenges, particularly in content moderation. One significant drawback is the absence of real-time content moderation, where harmful or inappropriate content can be detected and flagged as the blog is being created. Without this feature, harmful content may be published before it is reviewed, leaving the platform vulnerable to inappropriate or offensive posts.

Content moderation in traditional blog systems is usually handled manually by admins or moderators, which leads to several inefficiencies. Since this process is time-consuming, it is often inconsistent, depending heavily on the availability and attentiveness of human moderators. This approach also opens the possibility of human error, where some harmful content might slip through while other harmless posts may be unnecessarily flagged. Furthermore, as the volume of posts grows, it becomes increasingly difficult to manage, and the quality of moderation can suffer.

Additionally, traditional blog platforms typically offer limited user authentication options. Most systems rely solely on email/password-based logins, which are both outdated and prone to security concerns. Modern platforms, in contrast, integrate more secure and flexible authentication methods, such as Google login, phone-based authentication, and even multi-factor authentication. Without these options, users are often inconvenienced, and their accounts are potentially less secure.
Another issue with traditional platforms is the lack of an integrated harmful content warning system. Users can publish blog posts instantly without receiving any alerts or warnings about inappropriate language, violations of community guidelines, or the potential impact of their content. This lack of proactive moderation fails to prevent the publication of content that may cause harm or violate platform policies. Together, these limitations highlight the need for more modernized and automated content moderation solutions, such as AI-powered systems that can offer real-time detection and improved user authentication.

2.2 Proposed System

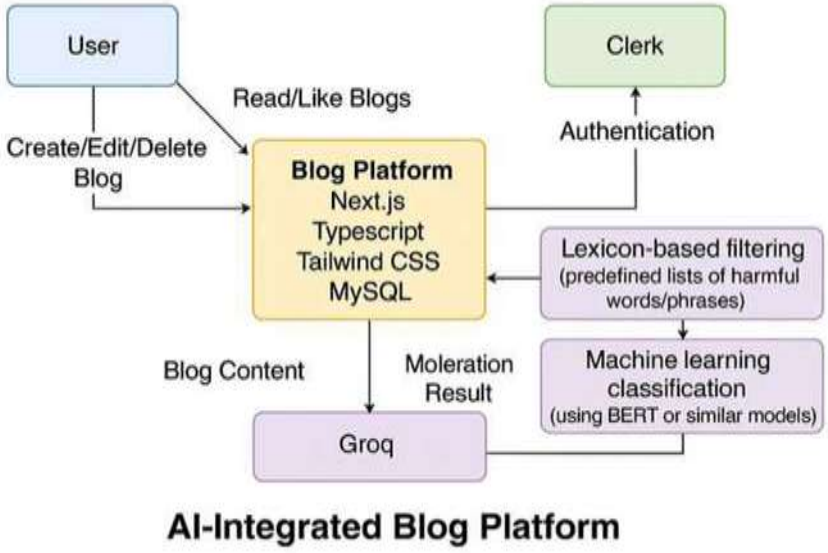

Fig. 1. Sequence Diagram

The proposed system introduces a significant upgrade to traditional blog platforms by incorporating cutting-edge technologies such as AI-powered content moderation, advanced user authentication, and enhanced user interaction features, creating a safer, more engaging, and efficient environment for users. One of the key innovations is the integration of AI-driven content moderation using Groq. This system enables real-time detection of harmful or inappropriate content before a blog post is published, ensuring that the platform remains free from offensive language, hate speech, or other violations of community standards. By automating this process, the proposed system addresses the shortcomings of traditional platforms that rely on manual moderation, significantly improving efficiency and reducing the risk of harmful content slipping through.
Real-time feedback during the blog creation process is another crucial component of the system. As users write their blog posts, the AI system continuously analyzes the content for potential violations and provides immediate alerts or suggestions to improve content quality and safety. This proactive approach not only prevents the publication of harmful material but also helps guide users toward creating content that aligns with the platform's guidelines. By intervening early in the creation process, the system fosters a more positive user experience and ensures that the content remains appropriate and valuable for the platform's broader community.
To further enhance the user experience, the proposed system integrates Clerk for advanced user authentication, offering multiple login options including email/password, Google login, and mobile

number authentication. This flexibility not only increases security but also improves accessibility, allowing users to sign in quickly and securely using their preferred method. The inclusion of modern authentication mechanisms also reduces the risks associated with outdated password-only systems, such as account hacking and unauthorized access, thereby enhancing overall platform security and user trust.

2.3 Module Description

The proposed blog platform is architected to seamlessly combine intelligent AI-powered content moderation with rich, user-centric blogging functionalities, establishing a dynamic yet safe space for online expression. This system is meticulously designed to uphold the standards of ethical and respectful digital communication, offering both creators and readers a secure and engaging environment. Unlike conventional blog platforms that often fall short in real-time content oversight, this solution introduces a sophisticated moderation mechanism without compromising the user experience.

At its foundation, the platform is built to be fully responsive, ensuring accessibility and seamless interaction across all devices including desktops, tablets, and mobile phones. This responsiveness is supported using Next.js, TypeScript, and Tailwind CSS, which not only enhance visual consistency and load performance but also contribute to a smooth user journey. The front end is tightly integrated with secure user authentication provided by Clerk, allowing for sign-ins via email/password, Google accounts, or phone numbers. This ensures flexible, secure, and personalized access to user dashboards, enhancing both convenience and data protection.

One of the standout aspects of the platform is its robust and modular AI moderation engine, which processes user-submitted content in real time through a three-phase filtering pipeline. The first layer—Lexicon-Based Filtering—performs rapid scanning of the text against a predefined list of harmful or inappropriate words. This initial pass eliminates blatantly offensive content with minimal processing time. The second phase leverages BERT-based classification, utilizing deep contextual understanding to evaluate phrases and detect more nuanced instances of toxic language, implicit threats, or hate speech that may not be captured through keyword matching. The final and most advanced layer, Contextual Analysis, employs transformer models such as LLaMA to interpret complex language patterns, including sarcasm, metaphorical expressions, and veiled abuse. This multi-layered moderation strategy ensures high precision in content filtering, reducing false positives and negatives, and maintaining the integrity of the platform's discourse.

In addition to moderation, the platform offers comprehensive blog management tools. Users can create, edit, save drafts, and publish blog posts through an intuitive rich-text editor that supports markdown, image uploads, and live previewing. As content is drafted, real-time moderation feedback is provided through unobtrusive alerts and suggestions, allowing users to revise and improve their posts before submission. This encourages proactive adherence to community guidelines and reduces the administrative burden on moderators.

From a system architecture perspective, the backend is developed using Node.js and MySQL, orchestrated through Express.js and integrated with Prisma ORM for type-safe, efficient database operations. Moderation services are abstracted into independent modules, enabling scalable deployments and streamlined maintenance. The backend supports RESTful APIs for seamless communication between components and is secured with token-based authentication to enforce access control and protect sensitive user data. For deployment consistency, the entire backend is containerized using Docker, with CI/CD support for automated testing and updates.

2.3.1 User Management Module

The User Management Module is a foundational component of the platform, facilitating secure, flexible, and seamless user authentication and profile handling through integration with Clerk. This module supports a wide range of sign-up and sign-in methods, including email and password, Google

authentication, and phone-based verification, allowing users to choose the option that best suits their preferences and needs. Once authenticated, users are granted access to a personalized dashboard, where they can manage their blog posts, monitor moderation feedback, and track the publication status of their content in real time.

This module emphasizes security and scalability, with Clerk providing robust session management that ensures a consistent user experience across multiple devices and browsers. Through encrypted sessions, automatic token refresh, and role-based access control (RBAC), Clerk enforces strict security standards while minimizing friction for users. These features work together to protect user identities and prevent unauthorized access, thereby enhancing the overall trust and safety of the platform.

Beyond basic authentication, Clerk also offers comprehensive profile management capabilities. Users can update their personal information, reset passwords, and manage linked authentication methods effortlessly from their dashboard. These features are powered by Clerk's well-documented and highly secure API, which simplifies backend logic while adhering to modern compliance frameworks.

In addition, Clerk's event-driven webhooks provide real-time interactivity by triggering platform-specific actions, such as sending email notifications, logging user activity, or updating moderation logs whenever a user signs in or makes profile changes. This capability enhances the platform's responsiveness and user engagement, enabling dynamic behavior that aligns with user interactions.

Overall, the User Management Module ensures that identity verification, session handling, and account management are tightly integrated and secure, providing a reliable foundation for the rest of the blogging platform. Its adaptability and built-in safeguards make it an essential component for maintaining both user convenience and operational integrity.

2.3.2 Blog Creation and Editing Module

The Blog Creation and Editing Module serves as a central component of the user experience, providing a comprehensive, intuitive, and feature-rich interface for composing and managing blog content. It is designed to empower users with the tools necessary to express their ideas while maintaining a seamless connection with the platform's intelligent moderation system. Built with responsiveness and accessibility in mind, the module ensures optimal functionality across all device types, including desktops, tablets, and smartphones.

At the heart of this module is a rich text editor that supports advanced formatting options, markdown syntax, and image uploads. This gives users the flexibility to create visually appealing and well-structured blog posts with ease. The editor also includes a cover image upload feature to enhance post aesthetics and encourage creative presentation. Upon submitting a post, the content is immediately routed to the AI moderation engine, where it undergoes automated analysis to ensure compliance with platform guidelines. This real-time integration guarantees that harmful or inappropriate content is flagged before publication, promoting a safe and respectful digital environment.

To improve content reliability and writing efficiency, the module incorporates autosave functionality, ensuring that user input is preserved continuously and reducing the risk of data loss due to unexpected interruptions. Users can also manage their work across different states—draft, published, or archived—offering flexibility for those who prefer to iterate and refine their posts over time. A real-time preview mode allows authors to visualize the layout and formatting of their content before finalizing it, improving clarity and user confidence during the publishing process.

Looking ahead, the module is designed to support collaborative features such as shared editing sessions, inline comments, and suggestion modes, making it suitable not only for individual creators but also for teams and group projects. These capabilities, though planned for future versions, will further enhance the platform's ability to foster collective knowledge sharing and high-quality content development.

By combining an intuitive writing interface with intelligent moderation and real-time feedback, the Blog Creation and Editing Module offers a responsive, interactive, and safe space for content

generation. It successfully balances freedom of expression with platform integrity, making it a core driver of user engagement and content quality in the proposed AI-powered blog platform.

2.3.3 AI Moderation Engine

The moderation engine serves as the central component of the platform, employing a robust three-phase filtering system to ensure the integrity and safety of user-generated content. The first phase, Lexicon-Based Filtering, involves the rapid scanning of submitted text against a predefined list of offensive or inappropriate words. This method enables the swift elimination of explicitly harmful content with minimal computational overhead.

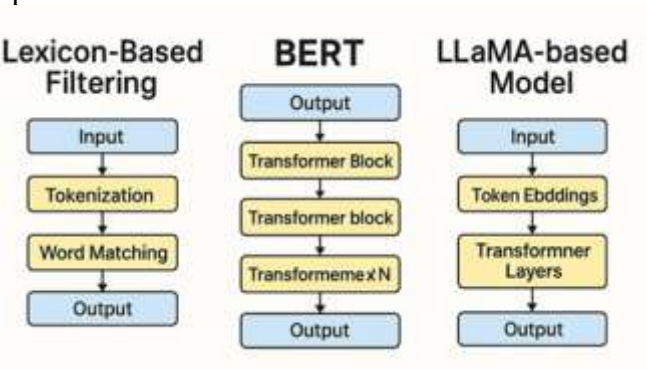

Fig. 2. Moderation Workflow Diagram

Once the content clears this initial screening, it advances to the BERT-Based Classification phase, where a BERT (Bidirectional Encoder Representations from Transformers) model conducts semantic analysis. This phase focuses on understanding the contextual meaning of the content, effectively identifying nuanced forms of hate speech, toxic language, or implicit threats that may not be detectable through simple keyword matching.

Finally, the content is processed through the Contextual Analysis Layer, which addresses more complex linguistic phenomena such as sarcasm, metaphorical expressions, and veiled abusive language. This layer utilizes advanced transformer-based models like LLaMA to assess intent and emotional tone, drastically reducing false negatives and ensuring that even subtle or disguised harmful content is accurately flagged or blocked. Collectively, this multi-tiered approach enhances moderation precision while maintaining a smooth and responsive user experience.

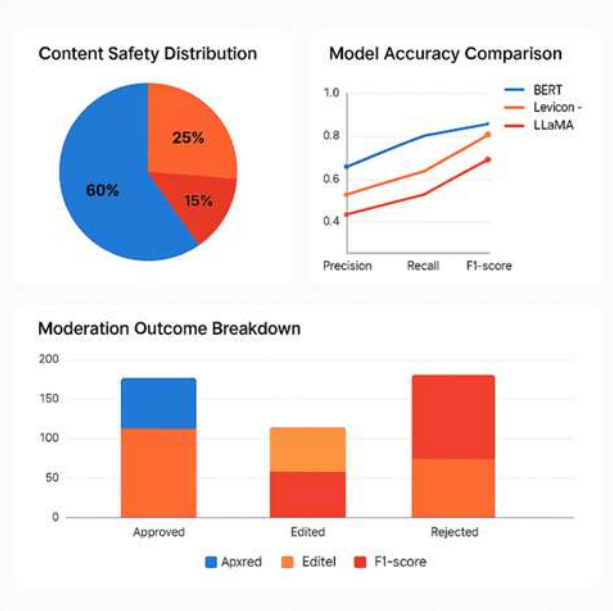

Fig. 3. AI Moderation Analytics Dashboard

The figure titled "AI Moderation Analytics Dashboard" provides a visual summary of the moderation system's performance across several key dimensions. The Content Safety Distribution pie chart in the top-left corner shows that 60% of content was directly approved, 25% required edits, and 15% was rejected for violating platform policies. This indicates that while a majority of user submissions are safe, a significant portion still needs intervention. In the top-right, the Model Accuracy Comparison line graph compares the performance of three moderation approaches—Lexicon-based filtering, BERT, and LLaMA—across precision, recall, and F1-score metrics. BERT leads in all categories, particularly in recall and F1-score, highlighting its strong capability in understanding context and subtle semantics. LLaMA shows better semantic handling than the lexicon approach, though not as effective as BERT. Finally, the Moderation Outcome Breakdown bar chart at the bottom illustrates how content was classified across the moderation pipeline. It categorizes posts into Approved, Edited, and Rejected, and further breaks each group down by model contribution. This combined visualization underscores the collaborative strength of the multi-model AI engine in delivering accurate and context-aware moderation outcomes.

2.3.4 Frontend Module

The Frontend Module of the platform is meticulously designed to provide a fast, responsive, and inclusive user experience, built with modern technologies that emphasize performance, accessibility, and ease of use. Developed using Next.js and TypeScript, the frontend benefits from powerful features such as server-side rendering (SSR) and static site generation (SSG), which contribute to faster load times, improved SEO, and enhanced performance across all devices. Styling is managed through Tailwind CSS, a utility-first framework that ensures clean, maintainable, and responsive layouts adaptable to various screen sizes using grid and flexbox systems.

Communication between the frontend and backend is securely handled using Axios, enabling efficient API calls for tasks like content moderation, user management, and blog operations. Authentication is seamlessly integrated using Clerk, allowing users to sign in via Google, email/password, or phone number, with persistent session support to maintain login states across sessions and devices. This improves both user convenience and platform security.

To foster real-time interactivity, the module includes toast notifications that provide instant feedback to users for critical actions such as blog submission, moderation outcomes, and system errors. This enhances the overall responsiveness of the interface and guides user behavior effectively. State management is implemented using React's context API and hooks, which track key application states including user authentication status, blog drafts, and moderation results, ensuring smooth transitions and dynamic UI updates.

The platform also adheres to modern accessibility standards, integrating features such as ARIA roles, keyboard navigation, and appropriate color contrast ratios. These practices ensure that the interface is usable and friendly for individuals with diverse abilities, contributing to a more inclusive digital environment.

Overall, the Frontend Module is not just a visual layer but a thoughtfully engineered system that bridges user interaction with intelligent backend services, offering a secure, adaptive, and high-performance interface that supports the platform's content moderation goals and user-centric design philosophy.

2.3.4 Backend Module

The Backend Module forms the operational backbone of the platform, designed to provide robust, scalable, and secure support for all critical functionalities, including blog content management, user data handling, and integration with AI-based moderation services. Developed using Node.js and MySQL, the backend is structured to deliver high-performance communication with the frontend while maintaining data consistency and integrity across the application.

At the architectural level, the backend follows RESTful API principles, enabling clear resource representation and efficient interaction between client and server components. The use of Express.js promotes a modular and organized codebase, simplifying request routing, middleware integration, and error handling. To streamline database operations, Prisma ORM is employed, providing a type-safe and intuitive interface for interacting with the MySQL database. This significantly reduces the chances of runtime errors and enhances both developer productivity and application stability.
A key architectural advantage lies in the decoupling of the moderation logic, which is implemented as a separate module. This separation allows the moderation engine to be independently deployed, tested, or scaled, depending on usage demand. Such a microservices-friendly design increases system flexibility, simplifies maintenance, and ensures efficient resource allocation.
Security is a major priority in the backend design. All API endpoints are protected through token-based authentication, ensuring that only authorized users and systems can access or modify sensitive data. This safeguards user sessions and supports role-based access control, aligning with modern application security best practices.
In addition to security, the backend is built for resilience. Comprehensive error handling, logging, and input validation mechanisms are integrated to detect and resolve issues proactively, preserving system uptime and improving debugging workflows. The entire backend stack is containerized using Docker, allowing for consistent behavior across development, staging, and production environments. This also simplifies deployment pipelines and team collaboration.
To support continuous delivery and quality assurance, the backend can be equipped with CI/CD pipelines, enabling automated testing, version control integration, and seamless deployment updates. These automation tools ensure that the backend remains reliable, maintainable, and adaptable to future feature expansion or performance enhancements.
In essence, the Backend Module serves as a well-architected, secure, and extensible infrastructure layer, enabling seamless blog management, efficient AI moderation, and a smooth user experience across the entire platform.

2.4 Technology Stack

The proposed blog platform is engineered using a modern, modular, and scalable technology stack, tailored to support rapid development, seamless integration, intelligent moderation, and an enhanced user experience. The architecture combines cutting-edge frontend and backend technologies with AI services, ensuring the platform is efficient, secure, and adaptable to evolving user and operational needs.
On the frontend, the system is built with Next.js and TypeScript, enabling the development of dynamic, type-safe interfaces that are both maintainable and scalable. These technologies are chosen not only for their performance and developer efficiency but also for their support of server-side rendering (SSR) and static site generation (SSG)—both essential for fast load times and SEO optimization. The UI is styled using Tailwind CSS, a utility-first framework that enables responsive and visually consistent designs across devices. API communication between frontend and backend services is handled via Axios, allowing secure, fast, and reliable data exchange. For user identity and access control, Clerk is integrated, offering multiple sign-in options including email/password, Google OAuth, and phone-based login, with seamless session management across sessions and devices.
The backend is developed using Node.js with Express.js as the framework to manage routing, middleware, and API logic in a structured and modular way. Data storage is managed by MySQL, a reliable and scalable relational database system, while Prisma ORM is used to facilitate type-safe and efficient interactions with the database. Prisma also improves productivity by abstracting complex queries and offering built-in data validation, which enhances the stability and security of backend logic.
A core innovation of the platform is its AI-powered moderation engine, which integrates the Groq API for AI inference and implements a layered moderation pipeline consisting of Lexicon-based filtering,

BERT-based semantic classification, and contextual analysis using transformer models like LLaMA. This sophisticated moderation strategy ensures real-time detection of toxic, harmful, or subtly inappropriate content, contributing to a safe and respectful environment for user-generated content.
In terms of development workflows, Git is used for version control, enabling collaborative development and continuous integration, while Postman supports API testing and validation throughout the development lifecycle. The platform is deployed using Vercel, a cloud platform that offers globally distributed hosting, automatic scaling, and performance optimization, particularly suited for Next.js applications. Containerization with Docker is used for consistent backend deployments, and CI/CD pipelines can be integrated to automate testing and streamline application updates.
Altogether, this technology stack reflects a thoughtful balance between developer productivity, user security, and application performance. It allows for real-time interaction, scalable moderation, and responsive user experiences—making the platform a robust and future-ready solution for intelligent content publishing.

III. Conclusion

The proposed AI-powered content moderation system offers a comprehensive, scalable, and forward-thinking solution for managing user-generated content in modern blog platforms. Unlike traditional moderation approaches that rely solely on manual review or simple keyword matching, this system employs a multi-layered AI-driven architecture that ensures both accuracy and efficiency in detecting harmful or inappropriate content. Its modular and API-driven design allows seamless integration into various web applications, making it adaptable to diverse development environments and technology stacks.
At the core of the system is a three-phase moderation engine that combines lexicon-based filtering, BERT-based semantic classification, and contextual analysis through advanced transformer models like LLaMA. This layered approach enhances the system's ability to detect not only explicit violations but also subtle forms of abuse, such as sarcasm or veiled threats—areas where traditional methods often fall short. By automating content analysis in real time, the system significantly reduces moderation delays while maintaining high standards of content integrity.
Another defining aspect of this solution is its focus on user experience and real-time feedback. As users compose blog posts, the platform provides live moderation alerts and suggestions, helping them align with community guidelines without disrupting the creative process. This proactive feedback loop promotes responsible content creation while reducing the burden on administrators.
Additionally, the system prioritizes security and accessibility through integration with Clerk for multi-method user authentication and session management, including support for Google login, email/password, and phone-based access. The frontend, built with Next.js, TypeScript, and Tailwind CSS, ensures responsiveness and accessibility across devices, while the Node.js and MySQL-based backend, containerized via Docker, guarantees reliability and scalability.
In summary, the proposed AI moderation platform combines intelligent automation, seamless integration, and user-centric features to deliver a safe, efficient, and inclusive blogging experience. It stands as a modern solution to the growing need for responsible content governance in digital communities.

References

[1] Firoj Alam, Shafiq Joty, and Muhammad Imran. "Domain-specific abusive language detection: A transfer learning approach." Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4860–4866, 2018.
[2] Ziqi Zhang, David Robinson, and Jonathan Tepper. "Detecting hate speech on Twitter using a convolution-GRU based deep neural network." The Semantic Web, Springer, pp. 745–760, 2018.

[3] Ye Zhang and Byron Wallace. "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification." arXiv preprint arXiv:1510.03820, 2015.
[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805, 2018.
[5] Taku Kudo and John Richardson. "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing." Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 66–71, 2018.
[6] Angela Fan, Shruti Bhosale, Holger Schwenk, et al. "Beyond English-centric Multilingual Machine Translation." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6024–6036, 2020.
[7] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature, 521(7553):436–444, 2015.
[8] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, et al. "Enhancing sentence embedding with generalized pooling." Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 578–583, 2018.
[9] Angela Fan, Edouard Grave, and Armand Joulin. "Reducing Transformer Depth on Demand with Structured Dropout." International Conference on Learning Representations (ICLR), 2020.
[10] John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. "Deeper attention to abusive user content moderation." Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1125–1130, 2017.