

Prime Pulse: An Enterprise Role-Based Web-Based News Content Management System

Chinmaya Mohanta
Student, Dept. of CSE,

Debendra Prasad Samal
Student, Dept. of CSE

Prof. Smruti Smaraki Sarangi
Guide, Dept. of CSE

GIFT Autonomous, Bhubaneswar *GIFT Autonomous, Bhubaneswar* *GIFT Autonomous, Bhubaneswar*

Abstract—The rapid expansion of digital media has created an urgent need for purpose-built, enterprise-grade content management platforms tailored specifically to newsroom operations. This paper presents Prime Pulse, a fully functional web-based news publishing portal that addresses the fragmentation and workflow inefficiencies prevalent in contemporary digital journalism ecosystems. The platform integrates a complete editorial lifecycle management system encompassing article creation, multi-level approval workflows, breaking news designation, scheduled publishing, and reader engagement management into a single unified system. Prime Pulse is built on a classic LAMP stack (PHP, Apache, MySQL) augmented with a granular Role-Based Access Control (RBAC) system governing four user roles — Administrator, Editor, Reporter, and Moderator — each with dedicated dashboards and middleware-enforced permission controls. A comprehensive analytics module delivers newsroom-specific performance metrics including category performance, per-reporter productivity rankings, peak traffic hour analysis, and breaking news versus regular news comparison. Security mechanisms include CSRF token protection, bcrypt-compatible password hashing, SQL injection prevention via prepared statements, MIME-type file validation, and rate limiting. A newsletter broadcasting system automatically notifies subscribers upon article publication without dependency on third-party services. Comprehensive testing across fifteen functional, security, and compatibility dimensions validates the platform's correctness, robustness, and readiness for production newsroom deployment. The project demonstrates that a feature-rich, enterprise-quality news content management system can be constructed entirely using open-source technologies at minimal operational cost.

Index Terms—Web-Based CMS, PHP, MySQL, Role-Based Access Control, News Publishing Portal, Editorial Workflow, Analytics Dashboard, CSRF Protection, Newsletter Broadcasting, LAMP Stack, Content Approval, Newsroom Management, RBAC, Agile Development, Scheduled Publishing.

I. INTRODUCTION

The proliferation of internet technologies and digital communication infrastructure has fundamentally restructured the global news media industry. News organisations of all sizes are transitioning away from print-centric production workflows toward digital-first publishing models that demand real-time content delivery, multi-channel audience engagement, and data-driven editorial decision-making. Despite this transformation, a significant structural gap persists: most small and medium-scale news organisations either rely on expensive proprietary CMS platforms or repurpose general-purpose content management tools that lack the editorial workflow, role-specific controls, and newsroom-centric analytics required for professional journalism operations.

General-purpose platforms such as WordPress, Drupal, Joomla, and Ghost have substantially advanced the field of digital content publishing. However, these platforms address isolated dimensions of newsroom operations rather than providing an integrated, end-to-end editorial management solution. Features critical to professional newsrooms — multi-level article approval workflows, granular role-based access control, breaking news designation, per-reporter performance analytics, integrated newsletter broadcasting, and revision history tracking — are either absent from core installations or require extensive paid plugin ecosystems, adding complexity, cost, and security vulnerabilities.

Prime Pulse is architected specifically to bridge this gap. It is a full-stack web-based news publishing portal built on the PHP, MySQL, HTML, CSS, and JavaScript technology stack that integrates all major editorial, analytical, and reader engagement functionalities into a single unified, secure, and cost-effective digital ecosystem. The platform supports four distinct staff roles — Administrator, Editor, Reporter, and Moderator — each with dedicated, purpose-built interfaces and middleware-enforced role-specific permission controls.

Primary Contributions: The contributions of this work are: (1) a complete LAMP-stack news publishing portal integrating editorial lifecycle management, reader engagement, and subscriber communication; (2) a granular four-role RBAC system with configurable per-permission toggle controls at runtime; (3) a newsroom-specific analytics dashboard providing breaking versus regular news comparison, peak hour traffic analysis, and per-reporter productivity rankings; (4) an integrated newsletter broadcasting system eliminating third-party service dependencies; and (5) a comprehensive security framework combining CSRF protection, prepared statements, MIME-type validation, and rate limiting.

The paper is structured as follows: Section II reviews relevant literature. Section III describes system design and architecture. Section IV details the technology stack. Section V presents implementation details. Section VI reports testing results. Section VII discusses findings and limitations. Section VIII presents conclusions and future research directions.

II. LITERATURE SURVEY

A. Evolution of News Content Management Systems

Web-based content management systems first emerged in the late 1990s as digital complements to traditional print publishing. Early platforms focused on static page management and basic article submission workflows. Over subsequent decades, these systems evolved to incorporate rich text editors, media libraries, comment management, and basic user role differentiation. Sommerville [1] and Pressman [2] establish the foundational software engineering principles — including requirements analysis, modular system design, and iterative testing methodologies — that underpin the development of enterprise-scale publishing platforms such as Prime Pulse.

Despite this historical evolution, empirical analysis of commercially available CMS platforms reveals persistent architectural gaps specifically for newsroom contexts. WordPress [3], the most widely deployed CMS globally, requires substantial plugin augmentation to support multi-level editorial approval, role-based content visibility, breaking news tagging, and analytics tailored to news content performance. These plugin dependencies introduce security vulnerabilities, performance overhead, and maintenance complexity that are impractical for resource-constrained news organisations.

B. Role-Based Access Control in Web Applications

Role-Based Access Control (RBAC), formalized by Sandhu et al. [4] in their landmark IEEE Computer publication, is the dominant access control paradigm in enterprise web applications. RBAC restricts system resource access based on predefined user roles and their associated permission sets, providing a structured, auditable, and maintainable alternative to discretionary access control. In multi-stakeholder news publishing contexts — where reporters, editors, moderators, and administrators must each interact with different content at different workflow stages — RBAC is architecturally essential.

Improper access control in news publishing systems creates serious operational risks: reporters directly publishing unreviewed content, moderators accessing confidential editorial discussions, or unauthorized users manipulating breaking news designations. Prime Pulse implements a four-role RBAC model enforced at the middleware layer of every admin panel access point, ensuring that no staff user can access resources or trigger workflow operations beyond their role-specific permission boundary.

C. Analytics in Content Management Platforms

Analytics integration in CMS platforms has evolved from basic page view counters toward sophisticated behavioural analytics infrastructures. General-purpose analytics tools such as Google Analytics provide broad traffic insights but lack the newsroom-specific metrics required for editorial decision-making: per-reporter article performance, category audience trends, breaking news audience capture rates, peak publication hour analysis, and editorial team productivity benchmarks. Pressman [2] identifies requirements specificity as a core determinant of software system quality, a principle that directly motivates Prime Pulse's dedicated newsroom analytics module over generic third-party integration.

D. Research Gap

Synthesizing the reviewed literature, a consistent structural gap emerges: no widely adopted open-source platform simultaneously provides complete editorial approval workflows, granular multi-role RBAC with runtime

configurability, integrated newsletter broadcasting, newsroom-specific analytics, and comprehensive security mechanisms within a single unified LAMP-stack system deployable at minimal cost. Prime Pulse is architected specifically to address this research and product gap.

III. SYSTEM DESIGN AND ARCHITECTURE

A. Overall System Architecture

Prime Pulse implements a classical three-tier web application architecture. The Presentation Tier is constructed with HTML5, CSS3, and Vanilla JavaScript, delivering a responsive, multi-interface experience for both public readers and authenticated staff. The Application Logic Tier is implemented using PHP 7.4+, managing session handling, business logic, workflow state management, email broadcasting via PHPMailer, and role-based permission enforcement. The Data Tier utilizes MySQL 8.0+ accessed exclusively through MySQLi prepared statements for security. The system operates on the Apache web server within an XAMPP development environment and is deployable on any standard LAMP hosting platform.

The system supports two primary interface contexts: a public-facing news website accessible to anonymous readers presenting breaking news, trending stories, category sections, and a photo of the day; and a secure admin panel accessible only to authenticated staff, encompassing article management, editorial workflow controls, analytics, subscriber management, role administration, and system configuration.

B. Authentication and Security Architecture

The authentication workflow employs a multi-layer security model. Admin user credentials are stored with PHP's password_hash() function using the COST-10 bcrypt algorithm, ensuring resistance to brute-force attacks even in the event of database compromise. Upon successful credential validation, a server-side session stores the authenticated user's ID, name, and role. A cryptographically random CSRF token is generated per session and embedded as a hidden field in every form submission, validated server-side before any state-modifying operation executes. The requireLogin() and requirePermission() middleware functions in the bootstrap file verify authentication status and role-based authorization on every admin panel request, with unauthorized access redirected to the login page.

Rate limiting is applied to login and public comment submission endpoints to prevent brute-force credential attacks and comment spam. All file uploads undergo MIME-type validation beyond extension checking to prevent malicious file disguise attacks. A JSON-based activity log records all administrative operations with user identity and timestamp for complete audit traceability.

C. Database Schema Design

The MySQL database, named news_portal, contains 19+ interrelated tables enforcing referential integrity through foreign key relationships. Table I summarizes the key collections and their primary fields. The news table is architecturally central, referencing categories, users, and being referenced by news_tags, comments, news_views, and news_logs tables. The role_permissions junction table creates a many-to-many relationship between roles and permissions, supporting runtime permission configuration without code changes.

TABLE I DATABASE SCHEMA — KEY TABLES AND PRIMARY FIELDS

Table Name	Primary Fields
news	id, title, slug, content, featured_image, author_id, category_id, status, is_breaking, is_top_story, views, scheduled_at, published_at, SEO fields
categories	id, name, slug, display_position
users / employees	id, username, password_hash, role_id, name, email, status, profile_photo
roles / permissions	id, role_name; id, permission_key
role_permissions	role_id (FK), permission_id (FK)
comments	id, news_id (FK), name, content, status (pending/approved/rejected)

Table Name	Primary Fields
subscribers	id, email, status, token, created_at
news_views	id, news_id (FK), ip_address, viewed_at
news_logs	id, news_id (FK), user_id (FK), action, created_at
notifications	id, user_id (FK), type, title, is_read, created_at

D. Editorial Workflow Design

The article editorial workflow follows a structured lifecycle enforced through database status transitions. A Reporter creates an article and saves it as a draft (status: draft). Upon submission, status transitions to pending, triggering both email and in-app notifications to Editors and Administrators. The Editor reviews the submitted article and may: approve it for immediate publication (status: published), reject it with written feedback (status: rejected, email notification sent to Reporter), schedule it for future publication (status: scheduled, cron-executed at specified datetime), or send it back for revision. Every status transition is recorded in the news_logs audit table with user identity and timestamp, providing complete editorial accountability.

IV. TECHNOLOGY STACK

The technology stack is selected to optimize for open-source accessibility, deployment simplicity, security, and long-term maintainability. The LAMP stack provides a universally supported runtime environment compatible with virtually all shared hosting providers, eliminating proprietary infrastructure dependencies. Table II provides a complete mapping of the stack components and their roles within the system.

TABLE II COMPLETE TECHNOLOGY STACK OVERVIEW

Layer	Technology	Version / Role
Backend	PHP	v7.4+ — Server-side scripting, business logic, session management
Database	MySQL	v8.0+ — Relational data storage with foreign key enforcement
Frontend	HTML5	Page structure and semantic markup for all interfaces
Frontend	CSS3 (Custom)	Responsive layouts for public website and admin panel
Frontend	JavaScript (ES6+)	Client-side interactivity, chart animations, UI enhancements
Charting	Chart.js	Latest CDN — Interactive analytics charts in admin dashboard
Email	PHPMailer	v6+ — Newsletter broadcasting, notifications, password resets
Server	Apache / XAMPP	v2.4+ — Local development and production deployment
Icons	Font Awesome	v6 CDN — Icon library throughout the admin interface
SEO	sitemap.xml / robots.txt	Static files for search engine indexing optimization

PHP's server-side execution model provides complete control over request handling, session management, and database interaction without exposing application logic to the client. MySQL's relational model with foreign key enforcement ensures data consistency across the 19+ table schema, supporting the complex JOIN queries required for analytics and reporting. Chart.js delivers interactive, animated analytics visualizations without requiring server-side rendering infrastructure. PHPMailer abstracts SMTP complexity, enabling both automated newsletter broadcasting and transactional email notifications through a unified API.

V. IMPLEMENTATION

A. Development Methodology

Prime Pulse follows an Incremental/Agile development model. Rather than constructing all modules simultaneously, the system was built in successive development cycles, each delivering a working, testable component. The first cycle established the database schema and authentication system. The second added article creation and the editorial approval workflow. Subsequent cycles introduced analytics, the newsletter system, role management, and the public-facing website. Each cycle followed the pattern: Plan → Design → Code → Test → Review, consistent with Agile methodology principles [5].

B. Public Website Module

The public website is the reader-facing interface of Prime Pulse. The homepage presents a responsive three-column newspaper-style layout: a trending stories sidebar on the left, a hero section featuring breaking and top news in the centre column, and a top stories sidebar on the right. A Photo of the Day section appears at the top of the main column. Breaking news articles are marked with a pulsing red indicator tag for immediate reader attention. The category pages provide paginated article listings; the article detail page features a full reading layout with gallery, embedded video, tags, and reader comments. Additional public pages include About, Contact, FAQs, Privacy Policy, Terms, and newsletter subscription management. SEO-optimised URLs using auto-generated slugs, per-article custom meta tags, sitemap.xml, and robots.txt improve search engine visibility.

C. Admin Panel and News Management Module

The admin panel, accessible at the /admin/ path, provides the complete newsroom management interface. The article creation page supports: title, auto-generated URL slug, category selection, short description, full rich-text content, featured image upload (JPG/PNG/WebP, max 2MB, MIME-type validated), gallery images, YouTube video URL embedding, SEO meta title and description, and tag selection from a searchable tag cloud. A sticky action bar provides role-appropriate buttons: Reporters see Save Draft and Send for Approval; Editors and Administrators additionally see Publish Now, Schedule, and Reject with Comment options. Article rejection triggers an automated email to the Reporter via PHPMailer containing the editor's feedback notes.

The admin panel encompasses: a dashboard with real-time newsroom metrics; article management (add, edit, manage, approve); category and tag administration; user and employee management; subscriber management; comment moderation; feedback management; notification centre; media library; photo of the day upload; analytics dashboard; roles and permissions manager; system settings; activity log; database backup; and revision history viewer.

D. Analytics Module

The analytics module (admin/analytics.php) delivers a comprehensive suite of newsroom performance metrics through interactive Chart.js visualisations. Metrics include: total articles and cumulative views, average views per article, today's published count and comment activity, unique visitor count (30-day rolling window), monthly growth trend chart (6-month history), 7-day traffic bar chart, content status distribution doughnut chart, peak traffic hours heatmap (24-hour distribution), content publishing velocity (8-week trend), top 10 most-viewed articles table, category performance rankings, per-reporter view leaderboard, popular tags word cloud, editorial activity summary, breaking versus regular news comparison cards, comment status breakdown, and CSV export for all data sets.

E. Security Implementation

Security is implemented at multiple independent layers. CSRF token protection is applied to every form submission: a cryptographically random token generated per session is validated server-side before any state-modifying database operation executes, preventing cross-site request forgery. All database interactions use MySQLi

prepared statements exclusively, eliminating SQL injection vulnerabilities regardless of input content. Uploaded files undergo dual validation: extension checking and server-side MIME-type inspection using PHP's `finfo` functions, preventing malicious executable files disguised with image extensions. Rate limiting is enforced on login attempts and public comment submissions using IP-based request counting stored in the sessions table. The storage/uploads/ directory is protected by `.htaccess` rules preventing direct browser access to uploaded files.

VI. SYSTEM TESTING AND RESULTS

A. Testing Strategy and Methodology

The Prime Pulse quality assurance framework is structured around five testing dimensions applied systematically across all platform modules. Functional testing verifies that each feature produces the specified output for both valid and invalid inputs. Security testing validates authentication enforcement, RBAC permission boundaries, CSRF token validation, and malicious file upload prevention. Performance testing measures public page load times and admin panel responsiveness under normal usage conditions. Integration testing confirms correct inter-module communication — specifically that article approval triggers both newsletter dispatch and in-app notifications. UI/Compatibility testing verifies responsive layout behaviour across Chrome, Firefox, Edge, and Safari on 320px (mobile), 768px (tablet), and 1280px+ (desktop) viewport widths.

Test cases were designed to cover each functional requirement defined in the System Analysis phase. Testing was performed systematically across all four authenticated staff roles (Administrator, Editor, Reporter, Moderator) and the unauthenticated Reader role to validate RBAC enforcement boundaries. All identified defects were logged, fixed, and re-verified before final release.

B. Functional Testing Outcomes

All fifteen functional test cases passed successfully across all modules and user roles. Table III summarises the functional testing outcomes.

TABLE III FUNCTIONAL TESTING SUMMARY

TC#	Test Case Description	Expected Result	Status	Role
TC01	Admin login with valid credentials	Redirected to admin dashboard	PASS	Admin
TC02	Admin login with invalid credentials	Error message; login denied	PASS	Admin
TC03	Reporter creates and submits article	Saved as pending; editor notified	PASS	Reporter
TC04	Editor approves a pending article	Article published; reporter notified	PASS	Editor
TC05	Editor rejects article with comment	Status rejected; reporter receives email	PASS	Editor
TC06	Admin schedules article for future	Status scheduled; auto-published at set time	PASS	Admin
TC07	CSRF token validation on form submit	Request rejected with security error	PASS	All
TC08	Malicious file upload prevention	Upload rejected; error message shown	PASS	Reporter
TC09	Reporter accesses user management	Access denied; redirected to dashboard	PASS	Reporter
TC10	Newsletter auto-send on article publish	All active subscribers receive email	PASS	Admin
TC11	Comment moderation workflow	Comment visible after moderator approval	PASS	Moderator

TC#	Test Case Description	Expected Result	Status	Role
TC12	Analytics view count accuracy	View count increments correctly	PASS	Admin
TC13	Article keyword search	Matching articles displayed correctly	PASS	Reader
TC14	Newsletter email subscription	Subscriber added; verification email sent	PASS	Reader
TC15	Role permission toggle update	Permission applied on next login	PASS	Admin

C. Security Testing Results

Security testing confirmed that CSRF token validation correctly rejects all form submissions with modified or missing tokens, returning an HTTP 403 response with no state modification. RBAC enforcement was validated by systematically attempting access to role-restricted admin panel pages using credentials from unauthorized roles; all unauthorized access attempts were correctly redirected to the dashboard with no data exposure. MIME-type image validation correctly rejected PHP scripts and executable files disguised with .jpg and .png extensions. Rate limiting on the login endpoint correctly blocked repeated credential attempts beyond the configured threshold. Database inspection confirmed that no plain-text passwords appear in the users table, with all credentials stored as bcrypt hashes. Cross-browser testing confirmed consistent rendering on Chrome, Firefox, and Edge; responsive layout testing confirmed correct behaviour across all three target viewport widths.

VII. DISCUSSION AND LIMITATIONS

A. Analysis of Outcomes

The implementation and testing outcomes collectively validate that Prime Pulse achieves its core architectural objectives. The LAMP stack provides a universally deployable, cost-free runtime environment confirming the architectural rationale for open-source technology selection. The four-role RBAC implementation proved particularly effective in enforcing workflow boundaries: all unauthorized access attempts were blocked during security testing with no data leakage or privilege escalation. The integrated newsletter system successfully eliminated third-party service dependencies, broadcasting publication notifications directly through PHPMailer's SMTP layer.

The analytics module delivered substantially richer newsroom-specific insights than general-purpose analytics integrations would provide, including per-reporter performance rankings, breaking versus regular news audience comparison, and publishing velocity trends that directly inform editorial strategy. The CSRF and prepared statement security layers, validated through systematic security testing, provide a robust defence-in-depth posture appropriate for production newsroom deployment. The comprehensive testing results — all fifteen functional test cases and all security test cases passing — confirm that Prime Pulse meets its full set of functional and non-functional requirements as specified in the System Analysis phase.

B. System Limitations

The current implementation has five primary limitations defining the boundary of the present contribution. First, Prime Pulse is web-only; no dedicated iOS or Android application exists for field journalists requiring mobile article authoring with offline capability. Second, the system does not support real-time simultaneous multi-user collaborative editing of the same article, limiting workflow efficiency when multiple editors need to contribute to a single piece. Third, scheduled publishing requires a server-side cron job to be manually configured to execute the `publish_scheduled.php` script; there is no built-in scheduler. Fourth, the article search implementation uses MySQL LIKE pattern matching, which does not support full-text relevance ranking or search suggestions. Fifth, published articles are not automatically shared to social media platforms, requiring manual distribution by staff.

VIII. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presented Prime Pulse, a fully functional enterprise web-based news content management system that addresses the fragmentation and workflow inefficiencies inherent in contemporary digital journalism operations. The platform successfully integrates a complete editorial lifecycle — from article creation and multi-level approval to publication, reader engagement, and performance analytics — into a single unified, secure, and cost-effective system built on the LAMP technology stack.

The multi-layered security architecture — CSRF token protection, bcrypt password hashing, SQL injection prevention via prepared statements, MIME-type file validation, rate limiting, and activity logging — comprehensively protects sensitive editorial data and enforces appropriate access boundaries across all staff roles. The four-role RBAC system with runtime-configurable permissions provides the granular access control required for professional newsroom operations without requiring code changes for permission adjustments. Comprehensive testing across fifteen functional modules, security scenarios, and browser compatibility dimensions validates the platform's reliability, correctness, and production readiness.

Prime Pulse establishes a reproducible, modular LAMP-stack architectural framework that development teams and researchers can adapt for modern digital journalism operations. The project demonstrates that a news management system comparable in functionality to expensive commercial platforms can be constructed entirely using freely available open-source technologies, making professional-grade digital publishing infrastructure accessible to news organisations of all sizes and resource levels.

B. Future Enhancement Directions

Four primary future enhancement directions are identified. First, development of native mobile applications for Android and iOS using React Native, enabling reporters to write, photograph, and submit articles from the field with native push notification delivery for breaking news alerts. Second, implementation of AI-assisted editorial tools integrating large language model APIs for article summarisation, SEO headline suggestions, automatic tag recommendations, and plagiarism detection, reducing manual editorial overhead. Third, full-text search enhancement using MySQL FULLTEXT indexing or Elasticsearch integration to provide relevance-ranked search results, phrase matching, and real-time search suggestions. Fourth, social media API integration enabling automatic article distribution to Twitter/X, Facebook, and LinkedIn upon publication, extending audience reach without additional staff effort.

Additional long-term research directions include: WebSocket-based real-time collaborative editing using Operational Transformation algorithms; multilingual CMS support using i18n frameworks for regional language news portals; microservices architectural decomposition for enterprise-scale deployment; a built-in PHP-based task scheduler replacing manual cron configuration; and a content monetisation module with subscription-based article paywalls and payment gateway integration for independent news organisations.

REFERENCES

-
- [1] I. Sommerville, *Software Engineering*, 10th ed., Pearson Education, 2016.
 - [2] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed., McGraw-Hill Education, 2019.
 - [3] WordPress Foundation, "WordPress Official Documentation," 2024. [Online]. Available: <https://wordpress.org/documentation/>
 - [4] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, Feb. 1996.
 - [5] K. Beck et al., *Manifesto for Agile Software Development*, 2001. [Online]. Available: <https://agilemanifesto.org/>
 - [6] PHP.net Official Documentation, "PHP Manual," 2024. [Online]. Available: <https://www.php.net/manual>
 - [7] MySQL Documentation, "MySQL 8.0 Reference Manual," Oracle Corporation, 2024. [Online]. Available: <https://dev.mysql.com/doc/>
 - [8] Chart.js Development Team, "Chart.js Documentation," 2024. [Online]. Available: <https://www.chartjs.org/docs/>
 - [9] PHPMailer Library, "PHPMailer on GitHub," 2024. [Online]. Available: <https://github.com/PHPMailer/PHPMailer>
 - [10] OWASP Foundation, "OWASP Top Ten Web Application Security Risks," 2024. [Online]. Available: <https://owasp.org/www-project-top-ten/>
 - [11] Google Developers, "SEO Starter Guide," 2024. [Online]. Available: <https://developers.google.com/search/docs/fundamentals/seo-starter-guide>
-

- [12] L. Welling and L. Thomson, PHP and MySQL Web Development, 5th ed., Addison-Wesley Professional, 2016.
- [13] R. Nixon, Learning PHP, MySQL & JavaScript, 6th ed., O'Reilly Media, 2021.
- [14] WordPress.org – Open Source Content Management Platform. [Online]. Available: <https://wordpress.org/>
- [15] Ghost Foundation – Modern Publishing Platform. [Online]. Available: <https://ghost.org/>
- [16] Drupal Association – Open Source CMS Platform. [Online]. Available: <https://www.drupal.org/>
- [17] Apache Software Foundation, "Apache HTTP Server Documentation," 2024. [Online]. Available: <https://httpd.apache.org/docs/>
- [18] Font Awesome, "Font Awesome Icon Library Documentation," 2024. [Online]. Available: <https://fontawesome.com/docs>