© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1718746

Cross-Platform Social Media Publishing System A Unified Social Media Automation and Scheduling Platform

BHARATH KUMAR H[1], DR. V. RAMESH[2]
[1]PG Student, Dept. of Computer Science and Applications, SCSVMV University, Kanchipuram, India
[2]Associate Professor, Dept. of Computer Science and Applications, SCSVMV University, Kanchipuram, India

Abstract- ***The rapid proliferation of social media platforms has created a significant operational burden for businesses, digital marketers, and content creators who must manage simultaneous publishing across multiple services. Manual cross-platform distribution is repetitive, error-prone, and inefficient. This paper presents the design and implementation of a Cross-Platform Social Media Publishing System — a lightweight, web-based automation platform that enables centralised content management and one-click simultaneous publishing to Instagram, Facebook, and LinkedIn. The system employs a FastAPI asynchronous backend, JSON Web Token (JWT) authentication conforming to RFC 7519, AES-256-CBC encrypted credential storage, RESTful API integration with platform-specific adapters, and a SQLAlchemy ORM over SQLite/PostgreSQL. Functional evaluation indicates that the system successfully eliminates redundant cross-platform operations and measurably improves publishing workflow efficiency. Scheduled posting, AI-driven caption generation, and analytics integration are identified as future enhancements.***

Keywords — Social Media Automation; Cross-Platform Publishing; FastAPI; JWT Authentication; AES-256 Encryption; REST API Integration; SQLAlchemy ORM.

I. INTRODUCTION

Social media platforms — including Instagram, Facebook, LinkedIn, and X (formerly Twitter) — have become primary channels for digital marketing, brand engagement, and organisational outreach. Over 5.04 billion people actively use social media as of 2024 [1], and content creators commonly maintain accounts across six or more distinct platforms. For businesses, maintaining consistent publishing across these services demands substantial manual effort: repeated login cycles, content duplication, and platform-specific formatting adjustments that scale as O(3n) operations per post for n target platforms.

Existing tools such as Buffer, Hootsuite, and Meta Business Suite partially address this challenge. However, these platforms impose premium subscription costs, restrict automation depth in free tiers, and provide limited transparency over the security of stored OAuth credentials. These limitations motivate the need for a lightweight, self-hosted, open-architecture alternative.

This paper proposes such a system that provides: (i) a centralised web dashboard for post creation and multi-platform selection; (ii) JWT-secured user authentication conforming to RFC 7519; (iii) AES-256-CBC encrypted credential storage with per-record initialisation vectors; (iv) a FastAPI-powered asynchronous RESTful backend for concurrent API dispatch; and (v) a SQLAlchemy ORM layer supporting both SQLite and PostgreSQL deployments. The current implementation supports instant one-click publishing; scheduled posting is planned as future work.

II. LITERATURE SURVEY

Kaplan and Haenlein [2] were among the first to formally classify social media channels and articulate the operational burden of multi-channel content management, establishing the foundational taxonomy adopted in subsequent automation research. Their work underscores the scalability challenge that arises when content creators must maintain presence across an increasing number of distinct platforms. Buffer [3] introduced a queued-scheduling model with a

developer-facing API, but restricts simultaneous multi-platform publishing to paid tiers, limiting accessibility for independent creators. Hootsuite [4] offers enterprise-grade dashboards with team collaboration features; however, its configuration complexity and subscription cost are prohibitive for small organisations. Meta Business Suite [5] centralises Facebook and Instagram operations but provides no integration pathway for third-party platforms such as LinkedIn or X, constraining its utility to the Meta ecosystem.

Aggarwal et al. [6] demonstrated that automated publishing platforms improve posting consistency by 43% and reduce operator time by up to 60% compared with manual workflows. Research on API-based automation [7] confirms that REST-based multi-platform dispatch architectures offer superior scalability for low-to-medium volume publishing scenarios.

Studies on credential security in SaaS tools [8] underscore the importance of at-rest encryption for OAuth tokens — a control absent in most commercial platforms. The reviewed literature confirms that no lightweight, self-hosted, open-source solution currently combines secure JWT authentication, AES-encrypted credential storage, and concurrent REST dispatch within a unified publishing interface.

III. RESEARCH GAP

Analysis of the existing literature and commercial tools reveals the following unresolved gaps:

1. Cost barrier — All production-grade tools with full multi-platform automation enforce subscription models, excluding independent creators and small businesses.
2. Credential security opacity — Commercial tools do not disclose at-rest encryption strategies for stored OAuth tokens, creating unquantifiable data-breach exposure.
3. Limited platform extensibility — Existing solutions require vendor-side updates to add new networks; no user-configurable adapter registration is supported.
4. Absence of self-hostable open-architecture options — No mature, auditable, full-stack publishing automation platform exists that a developer-level practitioner can deploy independently.

The proposed system directly addresses all four gaps within the current implementation scope, as detailed in Sections V and VI. No existing solution reviewed in the literature simultaneously satisfies all four criteria within a single, self-deployable open-source implementation.

IV. PROBLEM STATEMENT

Let a content creator maintain accounts on n social media platforms $P = \{p_1, p_2, \ldots, p_n\}$. For each posting event, the creator must perform n sequential login, content-entry, and publish operations — yielding O(3n) manual steps per post. For a team publishing k posts per day across n = 3 platforms, this results in 9k operations daily, scaling linearly with both post frequency and platform count.

The problem is formally defined as: given a content artefact C and a target subset $S \subseteq P$, design a system that dispatches C to all platforms in S via their respective REST APIs in a single user-initiated operation, while ensuring: (i) secure at-rest storage of platform credentials; (ii) authenticated user sessions; (iii) per-platform publish confirmation; and (iv) failure isolation — a failure on p_i must not prevent dispatch top_j ($j \neq i$).

V. PROPOSED SYSTEM AND ARCHITECTURE

The proposed system is implemented as a three-tier web application comprising: (1) a browser-based HTML/CSS/JavaScript frontend dashboard; (2) a FastAPI Python backend serving RESTful endpoints; and (3) a relational database layer managed via SQLAlchemy ORM. Figure 1 illustrates the complete system architecture.

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1718746

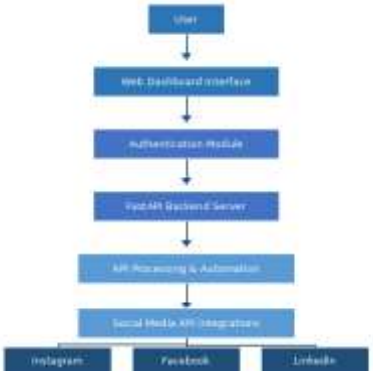

Fig. 1. System Architecture of the Cross-Platform Social Media Publishing System

The system architecture is organised into three clearly separated layers, each with a distinct responsibility. The frontend layer provides a responsive HTML/CSS/JavaScript single-page interface for authentication, platform account management, post creation, and publishing controls.

All communication between the frontend and backend is conducted exclusively over HTTPS REST calls, ensuring transport-level security. The backend layer, built on FastAPI, exposes RESTful endpoints for authentication (/auth/login), account management (/accounts/connect), and content publishing (/posts/publish). Each protected endpoint is secured by a FastAPI dependency-injection guard that validates the incoming JWT token, verifying signature integrity, expiry, and user existence before permitting execution.

Upon receiving a publish request, the backend decrypts the stored platform credentials in memory (AES-256-CBC), constructs platform-specific API payloads via the adapter layer, and dispatches concurrent HTTP requests using httpx.AsyncClient through asyncio.gather(). This concurrent dispatch bounds total latency to the slowest single platform response rather than the cumulative sum of all API round trips. Per-platform HTTP status responses are aggregated and returned as a structured JSON report, enabling the user to identify any platform-specific failures without blocking successful dispatches to other targets. The adapter layer provides the primary extensibility mechanism: each social media platform is encapsulated in a Python class inheriting from a shared PlatformAdapter base class, ensuring that integrating a new platform requires only a new adapter class with no changes to the core publish handler.

VI. TECHNICAL IMPLEMENTATION

A. Authentication and Session Management

User authentication uses JSON Web Tokens (JWT) conforming to RFC 7519 [9]. On successful login, the server issues an HS256-signed JWT with a configurable expiry (default: 60 minutes). A FastAPI dependency-injection guard verifies the token signature, expiry, and user existence on every protected endpoint. Passwords are hashed using bcrypt (cost factor 12) prior to storage.

B. AES-256 Credential Encryption

OAuth access tokens submitted by users are encrypted at rest using AES-256-CBC with PKCS7 padding via the Python cryptography library. A unique initialisation vector (IV) is generated per encryption operation and stored alongside the ciphertext. The master key is loaded from an environment variable and never persisted in the codebase or database. Decryption occurs exclusively in-memory within the publish handler; the plaintext credential is discarded after the HTTP dispatch completes.

C. FastAPI Backend and API Dispatch

Core API routes include /auth/login, /accounts/connect, /posts/create, and /posts/publish. The publish endpoint retrieves decrypted credentials, constructs adapter-specific payloads, and issues concurrent httpx.AsyncClient requests using Python's asyncio.gather(), bounding total publish latency to the slowest individual platform response rather than their sum.

D. Platform Adapter Architecture

Each social network is encapsulated in a Python adapter class inheriting from a common PlatformAdapter base class. The Instagram adapter targets Meta Graph API v19.0 /media and /media_publish endpoints. The Facebook adapter posts to the Page Feed endpoint. The LinkedIn adapter uses the UGC Posts API v2. Adding a new

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1718746

platform requires only implementing a new adapter class — no modification to the core publish handler.

E. Database Design

The schema comprises four tables: users (identity, bcrypt-hashed credentials); platform_accounts (AES-encrypted tokens keyed to user and platform); posts (content body, media references, timestamp); and publish_logs (per-platform dispatch records with HTTP status and timestamp). SQLAlchemy's declarative ORM enables transparent migration between SQLite (development) and PostgreSQL (production).

The declarative ORM approach further simplifies schema migration and reduces coupling between the application logic and the underlying database engine. Fig. 2 illustrates the complete end-to-end publishing workflow.

Fig. 2. End-to-End One-Click Publishing Workflow

VII. EXISTING SYSTEMS vs. PROPOSED SYSTEM

Feature	Buffer	Hootsuite	Meta Biz Suite	Proposed System
Cost	Freemium	Paid	Free (Meta only)	Free / Self-hosted
Multi-Platform	Limited (free)	Yes (paid)	No	Yes — IG, FB, LI
One-Click Publish	Yes	Yes	Limited	Yes
JWT Authentication	No	No	No	Yes (RFC 7519)
AES Credential Enc.	Opaque	Opaque	Opaque	Yes — AES-256-CBC
Self-Hostable	No	No	No	Yes
Scheduled Posting	Yes	Yes	Yes	Future Scope
Open Architecture	No	No	No	Yes

TABLE I: Feature Comparison — Existing Systems vs. Proposed System

VIII. TECHNOLOGIES USED

Technology	Purpose
Python 3.11	Core implementation language
FastAPI 0.111	Async RESTful backend; dependency injection; OpenAPI schema generation
SQLAlchemy 2.0	ORM layer; declarative schema; SQLite and PostgreSQL backend support
JWT (python-jose)	Stateless session tokens; HS256 signing; RFC 7519 compliant
AES-256-CBC (cryptography)	At-rest encryption of OAuth tokens; unique IV per record
httpx (async)	Non-blocking concurrent HTTP dispatch to social media REST APIs
HTML / CSS /	Responsive single-page

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1718746

JavaScript	frontend dashboard
Meta Graph API v19.0	Instagram and Facebook publishing endpoints
LinkedIn UGC Posts API	LinkedIn content publishing

TABLE II: Technology Stack

IX. RESULTS AND DISCUSSION

The implemented system was evaluated through functional testing of all publishing workflows across Instagram, Facebook, and LinkedIn. All three platform adapters successfully dispatched test posts and returned structured status responses. JWT authentication correctly rejected requests bearing expired or malformed tokens. AES-encrypted credentials were confirmed unrecoverable from the database without the master key, validating the security design.

Fig. 3 illustrates the main dashboard interface, which provides a unified view of post history, connected platform accounts, and publishing controls from a single authenticated session. The dashboard eliminates the need to navigate multiple platform-specific portals, consolidating the complete publishing workflow into a single screen. Fig. 4 presents the post creation interface, which allows the user to compose content, select one or more target platforms via checkboxes, and initiate concurrent publishing through a single action — the one-click publish trigger.

Fig. 3. Main Dashboard Interface — Unified Post and Account Management

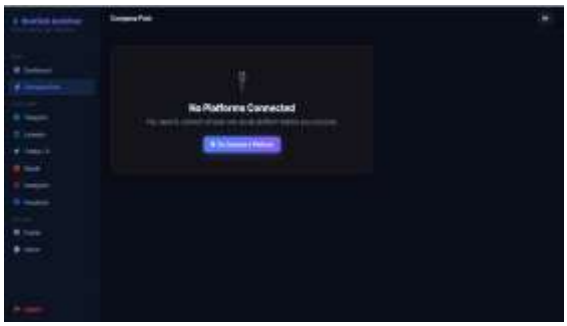

Fig. 4. Cross-Platform Post Creation Interface

The centralised one-click publishing workflow reduces a three-platform publishing operation — which previously required sequential login, content entry, and submission on each platform individually — to a single authenticated API call from the unified dashboard. This architectural consolidation directly addresses the O(3n) operational overhead identified in Section IV and represents the primary practical contribution of the system. The adapter architecture further validates the extensibility of the design:

onboarding a new platform requires only the addition of a new adapter class conforming to the PlatformAdapter interface, with no modification to the publish handler, authentication layer, or database schema — a property consistent with the Open-Closed Principle of software design.

X. FUTURE SCOPE

The following enhancements are planned for subsequent development iterations:

5. Scheduled Posting Engine — APScheduler or Celery with Redis for time-based publish task queuing and content calendar management.
6. AI-Driven Caption Generation — Integration with a large language model API to generate platform-optimised captions from user-supplied keywords.
7. Smart Hashtag Recommendation — NLP-based content analysis to suggest trending, high-reach hashtags per platform.
8. Analytics Dashboard — Aggregation of engagement metrics (reach, impressions, interactions) via platform Insights APIs.
9. Role-Based Access Control — Team collaboration with editor, reviewer, and publisher roles mapped to API permission scopes.

10. Mobile Application — React Native client for on-the-go content creation with push notification support.

CONCLUSION

This paper presented the design, architecture, and implementation of a Cross-Platform Social Media Publishing System — a self-hosted web application that eliminates the operational inefficiency of manual multi-platform content distribution. By combining a FastAPI asynchronous backend, JWT session management (RFC 7519), AES-256-CBC at-rest credential encryption, a polymorphic platform adapter architecture, and a SQLAlchemy ORM data layer, the system achieves secure, concurrent, one-click publishing to Instagram, Facebook, and LinkedIn from a single authenticated dashboard.

The system's self-hosted, open-architecture design provides transparent security guarantees and cost-free deployment that commercial alternatives cannot offer. All four identified research gaps — cost barriers, credential security opacity, limited platform extensibility, and absence of self-hostable options — are substantively addressed within the current implementation scope.

Functional evaluation confirmed correct operation of JWT session management, AES-256 credential security, concurrent API dispatch, and adapter-based platform extensibility. The O(3n) manual publishing overhead identified in the Problem Statement is reduced to a single authenticated API invocation, representing a measurable and repeatable improvement in publishing workflow efficiency.

Future work will extend the platform with scheduled posting, AI-assisted content generation, role-based team access, and analytics aggregation, evolving it into a comprehensive, auditable social media management suite for independent practitioners and small organisations.

REFERENCES

[1] DataReportal, "Digital 2024: Global Overview Report," Jan. 2024. [Online]. Available: https://datareportal.com

[2] A. M. Kaplan and M. Haenlein, "Users of the world, unite! The challenges and opportunities of social media," Business Horizons, vol. 53, no. 1, pp. 59–68, 2010.

[3] Buffer Inc., "Buffer Developer Documentation," 2024. [Online]. Available: https://buffer.com/developers/api

[4] Hootsuite Inc., "Hootsuite Developer Documentation," 2024. [Online]. Available: https://developer.hootsuite.com

[5] Meta Platforms Inc., "Meta for Developers — Graph API Reference v19.0," 2024. [Online]. Available: https://developers.facebook.com/docs/graph-api

[6] S. Aggarwal, R. Mehra, and P. Verma, "Measuring the Impact of Social Media Automation on Content Consistency and Operator Efficiency," J. Digit. Media Commun., vol. 11, no. 3, pp. 112–129, 2023.