International Journal of Innovation and Multidisciplinary Research
DOI.ORG/10.67505/IJIAMR.v6.i2.62004

Vol. 6 (2), 2026
E-ISSN: 2583-4452

SmartPost AI: An AI-Assisted Framework for Intelligent Professional Content Creation and Automated LinkedIn Management

Priya Sirohi[1], Shiksha Shrivastava[2*], Deepali Bajaj[3], Pushkar Gole[4]

[1] Swaja Robotics Pvt. Ltd., India

[2] Department of Computer Science, Faculty of Mathematical Sciences, University of Delhi, Delhi

[3, 4] Department of Computer Science, Shaheed Rajguru College of Applied Sciences for Women, University of Delhi, India

[*] Corresponding Author ✉ shikshashrivastava87@gmail.com

Abstract

The rapid development of Large Language Models (LLMs) has had a significant impact on the area of AI-assisted content generation, providing capabilities to generate coherent, context-aware text for different purposes. However, current solutions of AI writing assistants are limited in terms of addressing all aspects of professional content management, which include contextual information retrieval, organization, scheduling, persistent cloud storage, and automated publishing. Therefore, users are required to use several separate solutions, making their workflow less efficient and more labour-intensive. In this paper, we introduce SmartPost AI — an end-to-end framework for AI-assisted professional content creation and LinkedIn management. The framework utilizes a combination of Retrieval-Augmented Generation (RAG), LLMs, cloud-based data management, and LinkedIn API-based publishing, all integrated in one multi-layered architecture. The proposed framework enables users to provide either text prompts or web-based content, retrieve relevant contextual information, generate LinkedIn posts in a professional format, convert Markdown output into Unicode-based formatting, and manage content centrally via secure storage, scheduling, and publishing capabilities. The framework is deployed as a cloud-based full-stack web application using Next.js, FastAPI, Supabase, and LinkedIn APIs. Deployment and experimentation of the framework demonstrate that the AI-based content generation, workflow orchestration, cloud storage, and auto-publishing features have been successfully implemented within a single coherent framework. The suggested framework simplifies the process of content creation and makes it more automated, creating the basis for the intelligent AI-assisted content management and publishing system.

Keywords: *Large Language Models, Retrieval-Augmented Generation, AI-assisted Content Generation, LinkedIn Automation, Content Management, Workflow Automation, Cloud Computing*

1. Introduction

The fast development of Large Language Models (LLMs) has revolutionized the production of digital content by automating the process of creating meaningful, contextually relevant, and human-like texts for different applications (Feuerriegel et al., 2024). Social networks like LinkedIn have become vital means of building one's reputation, disseminating valuable information, and exchanging it with others.

Therefore, it is important for people and companies to consistently create high-quality, engaging, and professionally formatted content. Yet the process of creating such content is laborious and requires ideation, information collection, writing, formatting, hashtag creation, scheduling, and posting.

Recent advancements in generative AI have led to a wide array of AI writing assistants capable of creating professional text from natural-language input. Even though these tools have demonstrated strong proficiency in text generation, they offer limited support for the entire content management process. Many existing AI writing tools primarily rely on user input and offer limited integration of external knowledge, thereby increasing the likelihood of hallucinations and outdated information (Ji et al., 2023). In addition, many such tools provide limited support for the broader content-management workflow, including formatting, persistence, scheduling, and publishing. As a result, users must manually coordinate multiple independent tools to prepare and publish professional content.

To overcome these limitations, this paper proposes SmartPost AI, an AI-powered end-to-end framework for intelligent professional content creation and automated LinkedIn management. In particular, the framework combines RAG, LLMs, cloud-based data management, and automated publishing capabilities into a cohesive architecture for the entire content lifecycle (Klesel & Wittmann, 2025). The framework receives either plain-text prompts or URLs as inputs, uses a retrieval technique to enhance the context, uses an LLM to generate professionally formatted content, translates the output to platform-

Received on: 10 August 2026 | Accepted on: 02 October 2026 | Published Online on: 20 November 2026

© 2026 The Author(s). This article is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0).

compatible Unicode format, and manages storage, scheduling, and automated publishing using cloud-based services. Through this integration, SmartPost AI reduces human intervention while increasing the efficiency and usability of professional content creation.

In contrast to traditional AI writing assistants that focus on text generation alone, the framework offers a holistic solution that not only includes text generation but also other activities such as context enhancement, output formatting, centralized content management, scheduling, and publishing. The modular layered architecture also helps in independent development, maintenance, and scalability of individual components of the system while facilitating secure communication between internal services and external platforms. Overall, this design effectively facilitates professional content creation and management through a single intelligent framework.

The key contribution of SmartPost AI lies in integrating RAG-based contextual enrichment with AI-assisted content generation and the subsequent stages of professional content management, including scheduling and automated LinkedIn publishing, within a single modular framework. The main contributions of this paper are summarized as follows:

- AI-enabled end-to-end framework for professional content creation using a combination of RAG, LLMs, cloud persistence, and automated publishing.
- A modular layered architecture that unifies content acquisition, contextual enrichment, AI-assisted content generation, Unicode-based output formatting, and centralized content management within a scalable workflow.
- Unified content management process including draft management, scheduling, cloud persistence, metadata management, and secure communication with external publishing platforms.
- Automated workflow for professional content creation where text prompts or web-based information is converted into publication-ready LinkedIn content.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on AI-assisted content generation and automated publishing systems. Section 3 presents the proposed SmartPost AI framework and its methodology. Section 4 describes the experimental setup and system implementation. Finally, Section 5 concludes the paper and outlines future research directions.

2. Literature Review

The rapid development of Artificial Intelligence (AI), LLMs, and AI-assisted software applications has transformed the landscape of digital content creation and social media management. Recent studies demonstrate that AI systems can automate tasks such as content generation, sentiment analysis, recommendation engines, and workflow automation. The current tools on the market offer different combinations of capabilities, while an integrated framework that connects these capabilities to context-aware AI content generation remains less explicitly addressed.

The development of the transformer language model was a breakthrough in NLP. In 2017, Vaswani et al. introduced the transformer model, which uses a self-attention mechanism instead of recurrent neural networks. This architecture has become the foundation of modern large language models (LLMs). Based on the Transformer architecture, GPT-3 was introduced (T. Brown et al., 2020), and it has demonstrated zero-shot and few-shot learning capabilities across a wide variety of NLP tasks, without any task-specific training. GPT-3 showed great abilities in article writing, summarization, translation, and conversation (T. Brown et al., 2020; OpenAI, 2023).

The next-generation developments such as GPT-4, Gemini, Claude, and LLaMA have brought improvements in reasoning, multilingualism, factual accuracy, and contextual understanding (Feuerriegel et al., 2024; Gannon et al., 2017; Masalkhi et al., 2024; Team et al., 2023). These models have made significant advances in AI-assisted writing, producing coherent, contextually accurate professional content with minimal user input. Although the above-mentioned models provide high-quality content, they are intended solely for generating text and not for performing other downstream activities such as scheduling, publication, workflow management, and analytics. Modern research has extended the capabilities of LLMs beyond conversational systems toward intelligent workflow automation and AI-assisted software applications. Rather than functioning solely as standalone chatbots, modern LLM-based systems can be integrated with external tools, cloud services, databases, and application programming interfaces (APIs) to support complex tasks such as content generation, information retrieval, scheduling, and automated publishing. These capabilities enable AI-based systems to support various stages of the software development process, improving efficiency and reducing repetitive manual effort (Andrejevic, 2019; Duan & Wang, 2024; Klesel & Wittmann, 2025; Lathkar, 2023; Saeidi & Hollensen, 2024; Walczak-Skałecka & Staniszewska, 2025).

The development of frameworks such as LangChain (Mavroudis, 2024) has facilitated the development of AI-driven software applications, as these frameworks provide structured methods for connecting LLMs with external systems and data storage capabilities. With the help of frameworks, developers can now build applications that integrate NLP with software services to enable intelligent workflows. Modular software architectures have also enhanced the efficiency and adaptability of AI-powered applications by integrating content creation, cloud storage, analytics, and platform connectivity into a single software ecosystem (Devlin et al., 2019; Duan & Wang, 2024).

AI has also gained popularity as a solution for social media management, enhancing user engagement and marketing effectiveness. The existing frameworks, such as Buffer AI, Hootsuite AI, Jasper AI, and Copy.ai offer AI-powered solutions for content writing, post scheduling, hashtag generation, and campaign management. These commercial frameworks save time by

eliminating manual content writing. However, most of these frameworks operate independently, requiring users to manually move their content between frameworks to edit, schedule, publish, and analyse it.

Moreover, the content provided by these tools is not personalized since the content created by them is generic and does not consider the long-term branding strategy as well as past postings.

Finally, recent research works have led to the development of an AI-powered system that predicts user engagement, suggests the best time to post, and increases content visibility using ML algorithms. Although these approaches improve scheduling and content management, they often rely on separate software components that require users to manually coordinate content generation, editing, scheduling, publishing, and performance monitoring. Consequently, the overall workflow remains fragmented and requires significant user intervention.

Table 1: *Comparison of Existing AI Content Generation Systems*

System	AI Content	Scheduling	LinkedIn Integration	Analytics
ChatGPT	Available	Not Available	Not Available	Not Available
Jasper AI	Available	Limited	Not Available	Limited
Copy.ai	Available	Not Available	Not Available	Not Available
Buffer AI	Available	Available	Available	Available
Hootsuite AI	Available	Available	Available	Available
SmartPost AI (Proposed)	**Available**	**Available**	**Available**	**Available**

There are specific requirements for professional networking frameworks regarding the nature of the content to be published there. Posts on LinkedIn have to be professional, credible, readable, and relevant to their audiences. Recent research shows that LLMs can produce high-quality professional articles, technical blog posts, product announcements, summaries of research results, and educational content. The technique of prompt engineering enhances content personalization with the user's preferences, writing style, audience type, and knowledge of a certain subject. Although such solutions are now available, current AI writing tools are still mostly reactive. People must continuously provide prompts, edit outputs, choose images, schedule publications, and monitor performance. Hence, the process is not entirely coherent (Priyanshu et al., 2024; Sun et al., 2025). Table 1 presents a comparison of the existing AI-based content generation and social-media management solutions discussed in this study.

The literature review reveals several limitations in existing AI-powered content creation frameworks. Most solutions focus only on text generation, whereas a fully automated professional content management process is not covered by such systems. The user is required to organize activities of content editing, image selection, scheduling, publication, and analysis of interactions independently. While modern LLMs can create high-quality professional text, they cannot act independently and require constant human interaction; they also cannot manage a sequence of tasks without additional tools (Hayat et al., 2019; Moro et al., 2016).

Despite the availability of scheduling and analytics functionalities in existing social media management frameworks, these systems generally lack a unified architecture that seamlessly integrates AI-assisted content generation with content management and automated publishing. Therefore, the entire process of professional content management stays fragmented.

There have been numerous developments so far in AI-assisted content creation; however, the existing literature addresses various aspects of the content workflow, including text generation, scheduling, and analytics. The integration of RAG-based contextual enrichment with AI-assisted content generation and subsequent content-management stages, such as scheduling and automated LinkedIn publishing, remains less explicitly addressed within a unified framework. Therefore, a research gap remains in investigating an integrated AI framework that connects context-aware content generation with the broader professional content lifecycle (Feuerriegel et al., 2024; Ji et al., 2023).

To overcome these shortcomings, this paper proposes SmartPost AI, an AI-assisted framework for professional intelligent content creation and automated LinkedIn management. The proposed framework integrates LLMs, RAG, cloud-based data management, intelligent scheduling, LinkedIn API integration, analytics, and modern full-stack web technologies within a unified software architecture. Unlike conventional AI-assisted writing assistants that primarily focus on content generation, SmartPost AI supports the entire professional content life cycle, from context-aware content generation to scheduling, automatic publishing, and analysis of posted content (Klesel & Wittmann, 2025; Masse, 2011; Walker et al., 2024).

3. Proposed Methodology

The next section elaborates on the methodology used to develop SmartPost AI, an AI-enabled framework for automating professional content creation and publication. This framework combines RAG, LLMs, cloud-based data management, and automated publishing services into a single architecture that could cover the entire content lifecycle, including creation, contextualization, scheduling, and publishing.

To ensure modularity and scalability of the proposed SmartPost AI framework, it is designed with a layered architecture in which each functional unit communicates safely via services. The methodology comprises five interrelated elements: the system

architecture and user interaction process; the AI-based content creation process; content management and publishing; and the overall system workflow.

3.1 Overall System Architecture

The proposed SmartPost AI framework adopts a modular layered architecture that separates user interaction, workflow coordination, AI-assisted content generation, data persistence, and external platform integration into independent functional layers. This separation of responsibilities simplifies system maintenance while allowing individual components to be extended without affecting the overall framework. The overall architecture is illustrated in Figure 1.

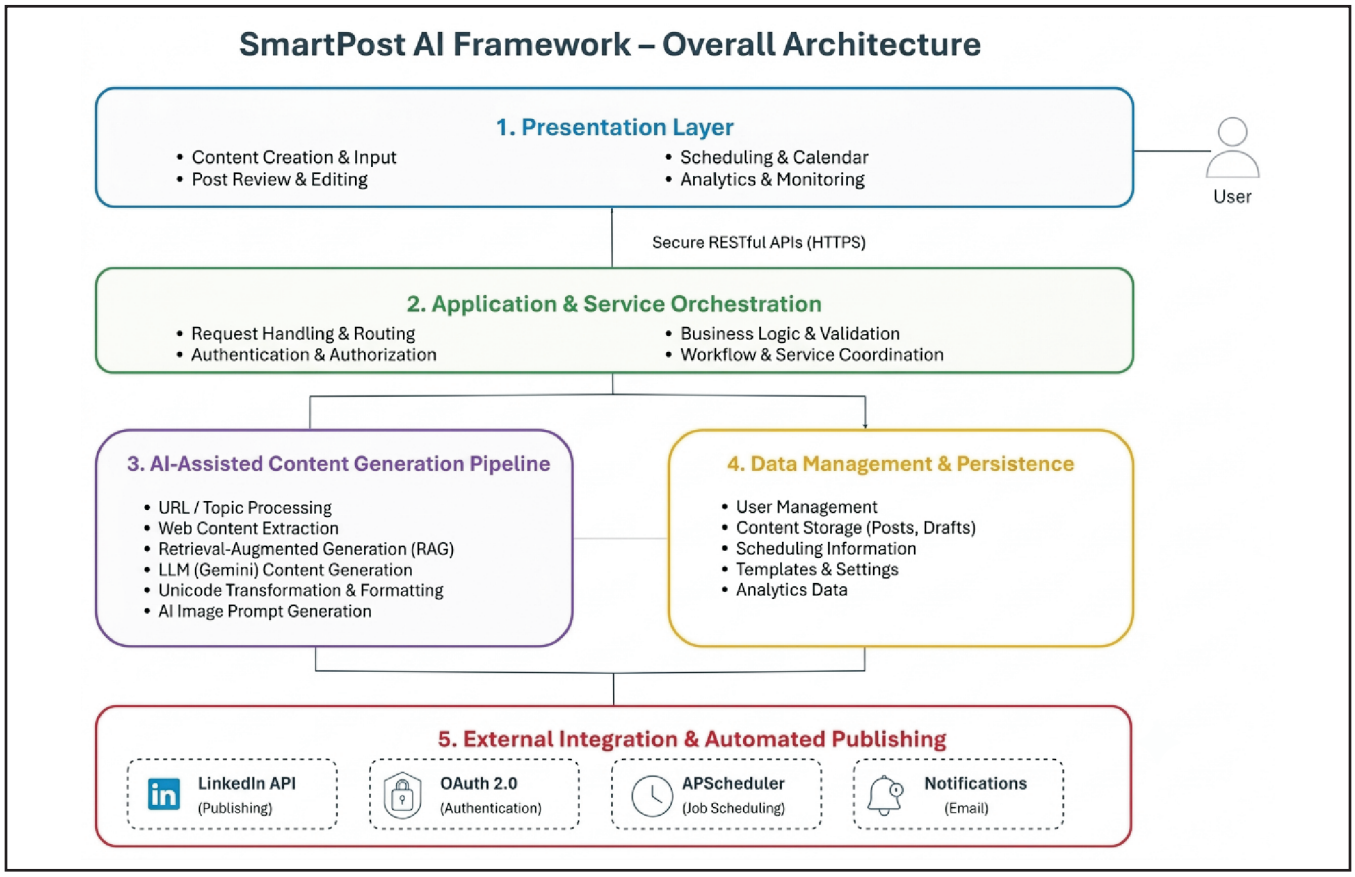


Figure 1: *Overall Architecture of SmartPost AI.*

As shown in Figure 1, the framework consists of five primary layers: Presentation Layer, Application and Service Orchestration, AI-assisted Content Generation Pipeline, Data Management and Persistence, and External Integration and Automated Publishing. Together, these layers help turn user input into professionally structured LinkedIn content while ensuring it can be securely stored, scheduled, and published.

The Presentation Layer provides the user interface for creating and managing content. The Application & Service Orchestration Layer handles user requests and facilitates communication between internal services. The AI-assisted Content Generation Pipeline acquires input data, performs RAG, generates AI-assisted content, and formats the output. The Data Management and Persistence Layer stores user information, created content, scheduling information, and metadata for publishing, whereas the External Integration and Automated Publishing Layer handles communication with external platforms to publish content.

3.2 User Interaction and Workflow Orchestration

The proposed operational workflow of the SmartPost AI solution framework begins when users interact with the system through the Presentation Layer, which serves as the interface for professional content

creation and management. Thus, users can create content using either textual prompts or web URLs, view and modify their posts, set up publication dates, and view their publishing status. User requests are validated and passed to the Application and Service Orchestration layer via RESTful APIs. This allows creating a clear distinction between the user interface and backend processes.

The Application and Service Orchestration layer coordinates all operations by dynamically routing requests to the appropriate processing modules. Requests using textual prompts are passed to the AI-assisted Content Generation Pipeline, while URL-based requests are processed by invoking the Content Acquisition module to obtain the context of the URL-based content. Other requests related to content management, including scheduling, publishing, and status checks, are routed to the respective

management services. During execution, intermediate results are passed between internal modules, aggregated, and sent back to the Presentation layer. This coordinated workflow enables efficient communication among independent system modules while supporting a modular and scalable architecture.

3.3 AI-Assisted Content Generation Pipeline

AI-assisted Content Generation Pipeline is the central part of the SmartPost AI framework, which serves to convert user inputs into socially relevant professional content. This pipeline consists of Content Acquisition, RAG, LLM-based content generation, and post-processing of generated content to produce factual, coherent content for professional social networking sites. The general flow of this pipeline is depicted in Figure 2.

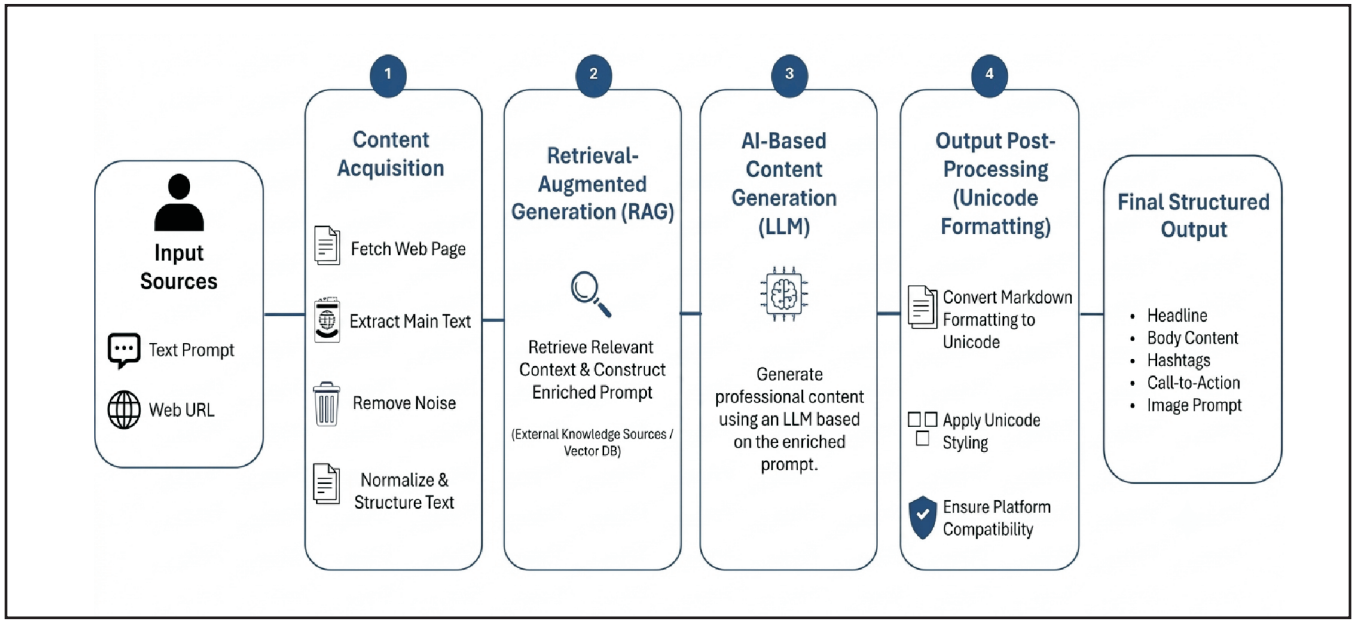


Figure 2: *Overall workflow of the AI-Assisted Content Generation Pipeline.*

3.3.1 Content Acquisition

The Content Acquisition phase prepares user data for future AI processing using two input types: textual prompts and external web URLs. The textual prompts are processed immediately, while URL-based inputs undergo web page retrieval to ensure their main textual content is captured properly. The use of this dual-input option enables the framework to generate professional texts based on either user-provided ideas or information from other web resources.

When it comes to URL-based input, irrelevant webpage components are also removed, such as navigation panels, ads, scripts, and style elements, before the relevant content is extracted from the website. This preprocessing step preserves meaningful contextual information while eliminating unnecessary noise, providing reliable input for the RAG stage.

3.3.2 Retrieval-Augmented Generation

After content acquisition, RAG is used to add relevant contextual knowledge to the processed input before generating the final response. While LLMs can only process the content of the user's prompt or the processed content of the web page, other information is also retrieved from external sources to increase consistency, completeness, and relevance. This combination of retrieved and processed content forms an enriched prompt, as defined in Eq. (1):

$$C_{aug} = \mathcal{P}\left(q, T_{clean}, \bigcup_{i=1}^{k} s_i, L, \tau\right) \tag{1}$$

where q denotes the user query, T_{clean} represents the cleaned input text obtained during Content Acquisition, s_i denotes the retrieved contextual snippets, L specifies the target language, τ represents the desired writing style or tone, and $P(\cdot)$ denotes the prompt construction function.

The enriched prompt helps the language model access additional contextual information beyond the original user prompt, enabling it to generate more consistent and contextually relevant professional content.

3.3.3 AI-Based Content Generation

The enrichment prompt is passed to an LLM that generates professional social media content based on the user's intent and context

gathered from external sources. By combining user intents with external knowledge, the LLM generates a consistent, coherent piece of content suitable for professional networking sites.

The generated content comprises various elements, such as a descriptive headline, body content, hashtags, a call to action, and image prompts that can be further generated by image-generation models. This process allows for the generation of all necessary parts in one go while maintaining stylistic consistency and minimizing manual preparation of the content.

3.3.4 Unicode Formatting and Output Preparation

Finally, output processing is performed to ensure that the content is compatible with professional social networking websites. Since many publishing websites only support basic Markdown formatting, the style information is converted to Unicode, retaining the same visual style. The formatting transformation is defined in Eq. (2):

$$f: M_{\text{Markdown}} \rightarrow U_{\text{Unicode}} \tag{2}$$

where M_{Markdown} denotes the generated Markdown-formatted content and $U_{Unicode}$ represents the corresponding Unicode representation suitable for platform-independent rendering.

Once formatted, the generated data is processed and arranged into a well-organized output format that contains the post content, hashtag, call-to-action, image prompt, and scheduling details. The well-structured data is then forwarded to the content management and publishing systems for scheduling and publishing.

3.4 Content Management and Publishing

After the creation of the AI-assisted content, the structurally formed output is transferred to the Content Management and Publishing stage, where it is stored, scheduled, and automatically published. The framework uses Supabase as the cloud backend service to store user profiles, generated posts, drafts, authentication, scheduling, and publishing data. Data centralization provides reliable storage and secure access to content. Generated content is stored with its metadata and may be either saved as a draft for further editing or scheduled for publication as needed. Generated posts are kept in historical records; thus, users can review their previous content, make edits, and republish without going through the process of generating new content. Also, authentication data, platform connection data, and scheduling data are stored to ensure the safe use of publishing services.

The generated content is published through the framework's integration with professional networking platforms via authenticated APIs. Depending on the selected publishing time, the content is either published immediately or scheduled for later publication. The publishing operation is represented by Eq. (3):

$$P_{out} = \begin{cases} \text{Publish}(C), & t = t_{\text{current}} \text{true} \\ \text{Schedule}(C, t), & t > t_{\text{current}} \end{cases} \tag{3}$$

where C denotes the generated content, t represents the user-defined publishing time, and P_{out} denotes the publishing operation. Once publishing is successful, the related status, timestamp, and platform-specific IDs are updated in the persistence layer. Unsuccessful publishing attempts are also logged to enable monitoring and future republishing. The integration of data management and automatic publishing in this way ensures an effective and robust solution for the entire lifecycle management of professional social media content.

3.5 End-to-End System Workflow

The workflow begins when a user submits either a textual prompt or an external web URL through the user interface. The process starts when the user submits a text prompt or a web URL via the UI. Following the validation of the request, the User Interaction and Workflow Orchestration process evaluates the appropriate execution path and passes the request to the AI-assisted Content Generation Pipeline.

For textual prompts, the input will be sent directly to the pipeline, while for URL inputs, the content acquisition and preprocessing stage will be performed before sending the input to the pipeline. Once the input is prepared, it is enriched using the Retrieval-Augmented Generation technique, which blends the available context with the retrieved data. Next, the LLM generates a professionally written LinkedIn post that includes a headline, body content, hashtags, a call to action, and an image prompt. Finally, the generated content is converted to a Unicode-compatible format to ensure consistent formatting across publishing platforms.

After content is generated, it is sent to the Content Management and Publishing component of the system, where it and its corresponding metadata are securely saved. Based on the user's choice, the content can be saved as a draft, scheduled for future posting, or published immediately, depending on the authenticated communication between the generated content and the destination platform. The result of the posting process is kept in sync with the persistence layer so that the user can check posting status, retrieve previously posted content, and plan future postings. This integrated workflow combines intelligent content generation, centralized content management, and automated publishing into a unified framework for professional LinkedIn content creation and management.

4. Experimental Implementation and Evaluation

This section describes the experiments conducted for the practical realization and quantification of the suggested SmartPost AI framework. As mentioned above, the system's multiple components make the evaluation process quite different from that of a

traditional machine learning model. Criteria such as the quality of content retrieval, the time required for this process, the quality of AI-generated content, the reliability and timing of content publishing, the effectiveness of scheduling, and the overall quality of process execution have been taken into account. In addition, a comparative evaluation is conducted between an LLM-only configuration and the proposed RAG-enhanced configuration to assess the contribution of contextual retrieval to the quality of content generation.

4.1 Development Environment

The implementation of the suggested SmartPost AI framework was presented as a cloud-based full-stack web application. The implementation of the system was quite consistent with the layered architecture of the proposed framework, ensuring that each functional layer operates independently while collaborating through well-defined service interfaces. This modular implementation strategy enhances software maintainability, supports future scalability, and enables efficient communication among user interfaces, backend services, AI components, cloud storage, and external platforms.



Figure 3: *Implementation of the user interaction and authentication interfaces: Onboarding landing portal, Supabase-based secured authentication interface, and Centralized workspace.*

The solution for data storage and user authentication has been achieved with Supabase, an open-source platform that provides PostgreSQL-based cloud storage and authentication services. All user information, content posted, scheduling information, publication information, and analytics are stored in the cloud database. Content generation is done using LLMs, and contextually relevant images are generated using Gemini 2.5 Flash. The connection to the LinkedIn platform is established using authenticated LinkedIn APIs.

The software technologies employed during implementation are summarized in **Table 2:**

Table 2: *Software Environment*

Component	Technology Used
Frontend	Next.js
Backend	FastAPI
Programming Language	Python, TypeScript
Database	Supabase (PostgreSQL)
Authentication	Supabase (Authentication)
AI Service	Large Language Model (LLMs)
Social Media Integration	LinkedIn API
Image Generation	Gemini 2.5 Flash
Version Control	Git & GitHub

4.2 Implementation of the Proposed Framework

The deployment of SmartPost AI follows the layered architecture proposed in Section 3. Each conceptual layer was realized as an independent software component responsible for a specific stage of the professional content management workflow. The chosen approach preserves modularity while ensuring effective communication among frontend services, backend processing, cloud storage, AI services, and external APIs.

The Presentation Layer has been implemented using Next.js, enabling users to interact with the application via the web interface to create professional content, review AI-generated answers, plan future publications, and view previous publications. User requests are communicated to the backend services via RESTful APIs.

The Application and Service Orchestration layer was developed utilizing FastAPI, which facilitates communication between AI services, cloud storage, authentication services, scheduling services, and the LinkedIn API. The asynchronous request processing enables multiple simultaneous service executions and improves the system's responsiveness.

The AI-Assisted Content Generation Workflow includes steps such as prompt preprocessing, RAG, LLM inference, Unicode formatting, and image generation to create publishable content on LinkedIn. This implementation reflects the conceptual design perfectly.

Supabase is the persistent storage component where user profiles, created content, publishing and scheduling details, and authentication information are stored. It is made possible by LinkedIn API integration and OAuth-based authentication. In general, the developed software framework effectively implements the conceptual model, demonstrating that intelligent content generation, workflow orchestration, cloud storage, and content publishing can be combined within a single framework.

4.3 Workflow Evaluation

The operational workflow of SmartPost AI was examined by executing the entire content generation and publishing process under normal working conditions. The process starts with authenticated users sending a content prompt using the website interface. The backend processes the user's request and passes the content prompt to the AI processing module to refine it and generate the content.

If the content is successfully generated, the visual content generation service is optionally invoked to produce images for the post. Then, the post is shown to the user to check the content, make changes if necessary, and publish it. Depending on the user preferences, the content can either be published directly or saved to the scheduling module to be published later.The content is automatically published when the schedule reaches the publishing time. At this moment, the publishing service interacts with the LinkedIn API to publish the content automatically. Once the content is successfully published, all metadata related to the generated post is saved to the cloud database, including the publication time, scheduling information, and user actions.

The complete execution pipeline showcases the operation of different components, such as artificial intelligence services, backend processes, cloud storage, and external social media APIs, working together within an integrated software architecture. The proposed workflow was then tested and analyzed in terms of retrieval performance, content-generation performance, reliability of publishing, scheduling performance, and execution performance.

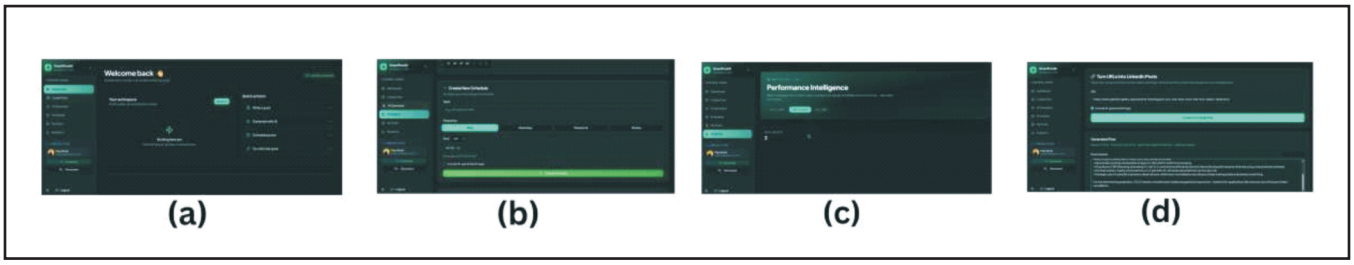


Figure 4: *Key features of the SmartPost AI web application include AI-assisted post creation, automated post scheduling, and performance analytics.*

The main interfaces of the designed system are shown in Figures 3 and 4 below, which include the Landing page, Registration and Login page, Dashboard, Intelligent content-generation interface, Scheduler interface, Analytics dashboard, and URL-to-post generation. This represents the practical implementation of the proposed architectural framework, integrating intelligent content generation, scheduling, publishing, and analytics into a single application.

4.4 Performance Evaluation and Analysis

The performance of the proposed SmartPost AI framework was analyzed using quantitative metrics such as retrieval performance, quality of generated content, publication performance, scheduling performance,

and execution of end-to-end workflow. The objective was to evaluate the main functionalities of the proposed system and obtain measurable data on its performance. Moreover, a comparative analysis between the system with an LLM-only setup and the proposed system with a retrieval-augmented generation (RAG) setup was conducted using the same set of prompts.

4.4.1 RAG Evaluation

The retrieval functionality was analyzed using twenty web search queries using the native retrieval process. The retrieval performance was measured using Precision@5 (Precision at cut-off 5), defined as the ratio of the number of top-5 items relevant to the query based on the topics and domain context. Retrieval time was measured for each query.

Results of the retrieval process are shown in Table 3 below. It can be observed that the system achieved a mean Precision@5 of 1.000, indicating that all evaluated queries returned relevant results among the top five items. The average retrieval time was 0.657 seconds.

4.4.2 LLM Content Evaluation

The content-generation module was evaluated using a test consisting of 20 LinkedIn posts generated by the system. In the experiment, a previously defined five-level Likert scale was used. This scale includes: 1= Very Poor; 2 = Poor; 3 = Acceptable; 4 = Good; 5 = Excellent. Four quality aspects were taken into account: relevance, factuality, coherence, and professionalism. Relevance included topic relevance, keyphrase relevance, and relevance to the input prompt. Factuality was evaluated by the

degree of consistency between facts and sources/references. Coherence was assessed using readability, paragraph structure, and list formatting, while professionalism was measured by LinkedIn language, hashtags, and calls to action.

To evaluate the generated content against the predefined criteria, a manual assessment was conducted, and the results are presented in Table 3 below. The generated content scored 4.27/5 for relevance and

3.54/5 for factuality. Coherence and professionalism received the maximal score of 5.00/5. The total quality score was 4.45 ± 0.23/5, while the average generation time was 8.325 seconds.

Table 3: *Evaluation of Retrieval Quality and AI-Generated Content*

Evaluation	Metric	Result
Retrieval	Precision@5	1.000
	Retrieval latency	0.657 s
Content generation	Relevance	4.27 ± 0.75 / 5
	Factuality	3.54 ± 0.66 / 5
	Coherence	5.00 / 5
	Professional tone	5.00 / 5
	Overall quality	4.45 ± 0.23 / 5
	Generation latency	8.325 s

4.4.3 Publishing, Scheduling, and End-to-End Evaluation

Publishing performance was evaluated through 20 real-time LinkedIn API publishing attempts. The publishing success rate and publishing latency were recorded for each attempt. Scheduling performance was evaluated using 10 scheduled-post requests by measuring registration latency and execution timing error. End-to-end performance was evaluated using 20 complete Prompt → RAG → LLM → Publish workflows under unthrottled conditions.

The publishing, scheduling, and end-to-end results are presented in Table 4. All 20 publishing attempts were successfully completed, resulting in a 100% publishing success rate. The mean publishing latency was 1.464 ± 0.521 s. For scheduled posts, the mean registration latency was 9.61 ± 4.91 ms, while the mean execution timing error was 28.4 ± 14.2 s. These results indicate that scheduling requests were registered with low overhead and that scheduled executions occurred with limited temporal deviation from their target times.

The end-to-end workflow also achieved a 100% success rate across the 20 evaluated workflows, with a mean processing latency of 10.45 ± 2.15 s. The end-to-end processing time is represented as:

$$T_{E2E} = T_{RAG} + T_{LLM} + T_{Publishing} \quad (4)$$

As defined in Eq. (4), the end-to-end processing time is the sum of the retrieval, content-generation, and publishing times. Based on the measured component latencies, the sequential processing time is approximately 0.66 + 8.33 + 1.46 = 10.45 s.

Table 4: *Publishing, Scheduling, and End-to-End Performance*

Evaluation	Metric	Result
Publishing	Success rate	100% (20/20)
	Publishing latency	1.464 ± 0.521 s
Scheduling	Registration latency	9.61 ± 4.91 ms
	Execution timing error	28.4 ± 14.2 s
End-to-end	Workflow success rate	100% (20/20)
	Processing latency	10.45 ± 2.15 s

4.4.4 Comparative Evaluation

To examine the contribution of retrieval augmentation to content-generation quality, a comparative evaluation was conducted between an LLM-only configuration and the proposed RAG + LLM configuration. Both configurations were evaluated using the same 20 prompts and the same quality-assessment procedure, enabling a direct comparison of the resulting content quality. The comparative results are presented in Table 5.

Table 5: *Comparative Evaluation of Content Quality*

Configuration	Mean Quality Score (1–5)	Relative Improvement
LLM-only	3.740	—
RAG + LLM	4.842	**29.47%**

The LLM-only configuration achieved a mean quality score of 3.740/5, whereas the RAG + LLM configuration achieved a mean score of 4.842/5. The RAG-enhanced configuration therefore achieved a 29.47% improvement in mean content quality relative to the LLM-only configuration. The improvement was calculated using the unrounded mean scores.

$$\text{Improvement}(\%) = \frac{Q_{\text{RAG+LLM}} - Q_{\text{LLM}}}{Q_{\text{LLM}}} \times 100 \quad (5)$$

Using the mean scores obtained in the comparative evaluation, Eq. (5) gives a relative improvement of 29.47% for the RAG + LLM configuration over the LLM-only configuration.

4.4.5 Overall Evaluation

Overall, the obtained quantitative results show that SmartPost AI successfully incorporates the basic steps of the AI-assisted professional content workflow. First, the results of the retrieval evaluation indicate consistent contextual retrieval; next, the evaluation of the content generation procedure shows very high content quality. The steps of the publishing process and the end-to-end workflow have been successful under the conditions considered, and the scheduling evaluation shows almost no registration overhead and low execution time deviation. In addition, the comparative evaluation results show that incorporating Retrieval-Augmented Generation (RAG) significantly increased the mean content quality score compared to the LLM-only approach.

The major interfaces of the proposed system are shown in Figures 3 and 4: Dashboard, Intelligent Content Generation Interface, Scheduler Interface, Analytics Dashboard, and LinkedIn Publishing Interface.

5. Conclusion and Future Work

This paper introduced **SmartPost AI**, a novel, AI-assisted framework for intelligent professional content creation and automation of LinkedIn operations. The proposed framework utilizes an approach that incorporates RAG, LLMs, cloud-based data management, Unicode-based output formatting, and LinkedIn API integration within a unified layered architecture to support complete professional content management. In this way, the problem of fragmentation in the processes of AI-assisted content creation is solved by combining the tasks of contextual content creation, centralized content management, scheduling, and automation of content publication into one workflow.

The proposed framework was implemented as a cloud-based full-stack web application using Next.js, FastAPI, Supabase, and LinkedIn APIs, following the modular architecture described in this work. Experimental implementation and functional evaluation demonstrated the feasibility of integrating AI-assisted content generation, workflow orchestration, cloud-based persistence, and automated publishing within a unified software framework. The modular design further provides scalability, maintainability, and flexibility for integrating additional AI services and extending the framework to evolving professional content management requirements.

However, some constraints are observed within the proposed framework, which is successful in demonstrating an exemplary model for applying AI assistants in professional content production and LinkedIn account management. In particular, the current design of the framework is specific to LinkedIn, relies on external LLMs and cloud services, and thus makes the framework's operation dependent on third-party APIs and network connectivity. The present evaluation is based on controlled technical experiments and does not include extensive user studies or long-term real-world deployment analysis.

The further development of the framework will be related to expanding its capabilities to cover the case of working with various social media platforms within the unified publishing framework, with such features as personalized content creation according to users' preferences and history of previous publications, predictive analytics scheduling, and multimedia content creation. Future work will also consider large-scale quantitative evaluation and assessment of practical effectiveness in real-world professional content management environments.

References

1. Andrejevic, M. (2019). *Automated media*. Routledge.
2. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., & others. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901.
3. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186. https://doi.org/10.18653/v1/N19-1423
4. Duan, Z., & Wang, J. (2024). Exploration of LLM multi-agent application implementation based on LangGraph + CrewAI. *ArXiv Preprint arXiv:2411.18241*.

5. Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative AI. *Business and Information Systems Engineering, 66*(1), 111–126. https://doi.org/10.1007/s12599-023-00834-7
6. Gannon, D., Barga, R., & Sundaresan, N. (2017). Cloud-native applications. *IEEE Cloud Computing, 4*(5), 16–21.
7. Hayat, M. K., Daud, A., Alshdadi, A. A., Banjar, A., Abbasi, R. A., Bao, Y., & Dawood, H. (2019). Towards deep learning prospects: Insights for social media analytics. *IEEE Access, 7*, 36958–36979. https://doi.org/10.1109/ACCESS.2019.2905101
8. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys, 55*(12). https://doi.org/10.1145/3571730
9. Klesel, M., & Wittmann, H. F. (2025). Retrieval-Augmented Generation (RAG). *Business and Information Systems Engineering, 67*(4), 551–561. https://doi.org/10.1007/s12599-025-00945-3
10. Lathkar, M. (2023). *High-Performance Web Apps with FastAPI*. Apress Berkeley.
11. Masalkhi, M., Ong, J., Waisberg, E., Zaman, N., Sarker, P., Lee, A. G., & Tavakkoli, A. (2024). A side-by-side evaluation of Llama 2 by Meta with ChatGPT and its application in ophthalmology. *Eye, 38*(10), 1789–1792.
12. Masse, M. (2011). *REST API design rulebook: Designing consistent RESTful web service interfaces*. O'Reilly Media.
13. Mavroudis, V. (2024). *LangChain v0.3*. Preprints, 2024, 2024110566. https://doi.org/10.20944/preprints202411.0566.v1
14. Moro, S., Rita, P., & Vala, B. (2016). Predicting social media performance metrics and evaluation of the impact on brand building: A data mining approach. *Journal of Business Research, 69*(9), 3341–3351. https://doi.org/10.1016/j.jbusres.2016.02.010
15. OpenAI, R. (2023). GPT-4 technical report. *ArXiv:2303.08774*.
16. Priyanshu, A., Maurya, Y., & Hong, Z. (2024). AI governance and accountability: An analysis of Anthropic's Claude. *ArXiv Preprint arXiv:2407.01557*.
17. Saeidi, S., & Hollensen, S. (2024). Digital marketing on LinkedIn: In-depth strategies for lead generation. *American Journal of Industrial and Business Management, 14*(5), 655–668. https://doi.org/10.4236/ajibm.2024.145033
18. Sun, Z., Zhang, Z., Shen, X., Zhang, Z., Liu, Y., Backes, M., Zhang, Y., & He, X. (2025). Are we in the AI-generated text world already? Quantifying and monitoring AIGT on social media. *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 22975–23005.
19. Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., & others. (2023). Gemini: A family of highly capable multimodal models. *ArXiv Preprint arXiv:2312.11805*.
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*.
21. Walczak-Skałecka, A., & Staniszewska, Z. (2025). Artificial intelligence and the CEO's personal brand: Opportunities, risks, and paradoxes. *Scientific Papers of Silesian University of Technology. Organization and Management Series*, (224), 597–619. https://doi.org/10.29119/1641-3466.2025.224.33
22. Walker, C., Gharaibeh, T., Alsmadi, R., Hall, C., & Baggili, I. (2024). Forensic analysis of artifacts from Microsoft's multi-agent LLM platform AutoGen. *Proceedings of the 19th International Conference on Availability, Reliability and Security*, 1–9.

❖ ❖ ❖