© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1711882

AI Social Media Manager

UTKARSH BARBHAI[1], SOHAM DHAPATE[2], HARSHAD POTDAR[3], SARVESH MOHITE[4]
[1,2,3,4] *Computer Science Engineering, MIT Art, Design & Technology University Pune, India*

Abstract*—The management of social media for brands and individuals is a critical but highly inefficient process. Professionals rely on a fragmented suite of tools for content creation, scheduling, and performance analysis , a disconnection that hinders the ability to translate analytical data into effective content strategy. This paper introduces the AI-Powered Social Media Manager, a unified platform designed to solve these challenges. The system's core contribution is a novel, stateful agentic workflow built with langgraph, which dynamically manages the entire content lifecycle. This architecture integrates a generative AI for image and text creation , an automated scheduling engine powered by apscheduler , and a comprehensive analytics dashboard. By creating an integrated feedback loop, the system analyzes post engagement and current trends to provide data-driven strategic recommendations for future content. Results demonstrate that this unified system significantly streamlines the management workflow, reducing the time-to-post from over 8 minutes to under 3. The successful implementation of this proof-of-concept demonstrates that unifying these core functions through an agentic AI represents a notable improvement over traditional, multi-platform management methods.

***Keywords*—Social Media Management, Generative AI, Agentic Workflow, LangGraph, Workflow Automation, Social Media Analytics, Content Strategy.**

I. INTRODUCTION

In the modern digital economy, a dynamic social media presence is a non-negotiable asset for businesses and content creators. With over 5.24 billion active users globally and a market size for social media management (SMM) software valued at over $24.47 billion in 2024, these platforms have become the primary channel for marketing, service delivery, and audience engagement.

However, the effectiveness of this presence is throttled by a workflow that remains fundamentally fragmented and labor-intensive. A typical social media manager's day is characterized by constant context-switching between disparate, single-function tools. The workflow is siloed:

1. Ideation & Creation: Content is conceived and then manually created in graphic design platforms like Canva or standalone generative AI tools (e.g., DALL-E, Gemini).
2. Scheduling & Publishing: This finished content is then downloaded and re-uploaded into a separate scheduling application, such as Hootsuite or Buffer.
3. Analysis: Performance metrics are tracked in yet *another* dashboard, either on the native platform or a third-party analytics tool.

This disconnection is not just a minor inconvenience; it represents a significant drain on resources, with studies indicating that inefficient business processes can cost companies 20-30% of their annual revenue. The most critical "gap" in this workflow is the lack of an automated, intelligent feedback loop. The insights gleaned from analytics (Step 3) are manually interpreted and must then be *remembered* and *translated* by a human operator back into (Step 1). This critical disconnect hinders the ability to translate analytical data into effective content strategy.

While the SMM market has identified the "integration of advanced analytics and AI" as a key driver, most existing solutions fail to unify all three functions. To overcome these challenges, we have developed the AI-Powered Social Media Manager. This paper introduces a unified, high-performance platform that consolidates the entire lifecycle of a social media post, from generation to analysis, into a single, cohesive system.

The primary contributions of this work are:

- A Unified API Backend: The design and implementation of a single, asynchronous API using fastapi that serves as the central hub for the user interface, AI engine, and database.
- An Agentic AI Core: The development of a novel, stateful agentic workflow using langgraph to dynamically route user prompts and manage the content lifecycle. This engine leverages high-speed LLMs via the groq API for real-time text and prompt generation.
- A Persistent, Automated Scheduler: The integration of a robust, background scheduling

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1711882

module using apscheduler to manage and publish content at precise, user-defined times.

- An Integrated Feedback Loop: The creation of an analytics-to-recommendation pipeline, using pandas to process engagement data and feed strategic insights directly back into the content generation agent.

II. RELATED WORK

The problem of social media management exists at the intersection of three distinct domains: traditional SMM platforms, standalone generative AI tools, and the underlying software architectures that can integrate them. Our review of the literature focuses on these three areas to identify the critical research gap.

A. Commercial Social Media Management (SMM) Platforms

The SMM platform market is mature, dominated by established players like Hootsuite, Sprout Social, and Buffer. These platforms are highly optimized for scheduling, analytics, and team collaboration. Their primary value lies in providing a unified dashboard for publishing content across multiple channels and aggregating performance metrics to demonstrate ROI.

However, these tools were architected before the advent of high-fidelity generative AI. Their recent integrations of AI are often limited to text-based assistance, such as AI caption writers (e.g., Hootsuite's OwlyWriterAI or Buffer's AI Assistant). These features are typically wrappers for large language models (LLMs) like GPT. They do not possess native, high-quality *image generation* capabilities and thus still require the user to perform the fragmented workflow of creating visual assets in an external tool, downloading them, and re-uploading them for scheduling. While Sprout Social has moved toward deeper AI integration for analytics and listening, the fundamental workflow of *generation* remains disconnected from *analysis*.

B. Standalone Generative AI in Content Creation

Concurrently, a separate revolution has occurred in content creation, led by generative models like OpenAI's DALL-E, Google's Gemini, and open-source models like Stable Diffusion. These tools offer unprecedented power to marketers, automating the creation of novel text and images and drastically reducing the time and cost of content production. Academic research has explored their use for everything from brainstorming to generating entire marketing campaigns.

The limitation of these tools is that they are "creative black boxes," completely disconnected from the strategic SMM lifecycle. They lack any scheduling or analytics functions. The content they produce is not informed by real-time performance data, and the models themselves have no awareness of the user's broader content strategy, audience engagement, or brand voice. This forces the user to manually bridge the cognitive gap, translating analytic insights into effective text prompts.

C. Agentic Workflows for System Integration

To bridge the gap between these two siloed domains, a new architectural paradigm is required. Traditional, linear software pipelines (often built with tools like LangChain) are insufficient, as they follow a rigid, predefined sequence of steps and cannot easily loop back on themselves. A social media "feedback loop" (where analytics inform generation) is, by nature, a cyclical, stateful process.

This is where agentic workflows, built using frameworks like langgraph, present a novel solution. Unlike simple chains, langgraph is designed to build stateful, multi-agent systems where workflows are represented as graphs. This allows for complex, dynamic behavior, such as a "router agent" that delegates tasks to specialist agents. Research in multi-agent systems highlights their power in solving complex, multi-step business problems by breaking them down into collaborative tasks. Our choice of langgraph is a direct response to the limitations of linear chains, enabling us to build a robust system that can maintain state, make conditional decisions, and, most importantly, create the feedback loop that is missing from all current commercial tools.

D. The Identified Research Gap

The literature reveals a clear dichotomy:

1. SMM platforms excel at distribution and analytics but fail at integrated, high-quality generation.

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1711882

2. GenAI tools excel at generation but fail at distribution and analytics.

The critical, unaddressed research gap is the creation of a single, unified platform that closes this loop. Our work is the first, to our knowledge, to propose and build a system that (1) integrates generation, scheduling, and analytics, and (2) uses a stateful, langgraph-based agentic workflow to create an intelligent feedback loop, allowing performance data to *autonomously* inform and improve future content generation.

III. SYSTEM ARCHITECTURE AND IMPLEMENTATION

A. High-Level Architecture

The AI-Powered Social Media Manager is designed using a modular, web-based architecture to provide a seamless user experience. The system's workflow, illustrated in Fig. 1, integrates all core functions from content creation to performance analysis into a single, cohesive platform. This architecture is centered around a backend API server that orchestrates interactions between the user interface, a core AI engine, a database, a task scheduler, and external social media services. This design ensures that the entire process, from a user's initial prompt to the final analytics display, is fully automated and efficient.

This architecture is centered around a backend API server that orchestrates interactions between the user interface, a core AI engine, a database, a task scheduler, and external social media services. This orchestration, as depicted in the diagram, follows a specific, automated 10-step process:

1. A user submits a request via the User Interface.
2. The API Server (FastAPI) receives the request and routes it to the AI Engine for processing (Step 2: "Process").
3. The AI Engine (LangGraph) generates the content (Step 3: "Generate Content") by calling External Services (like Groq API & GenAI Models).
4. The generated content and schedule are sent to the Database for persistence (Step 4: "Store & Schedule").
5. The API Server also registers the task with the Scheduler (Step 5: "Set Schedule").
6. At the scheduled time, the Scheduler triggers the API Server to post the content (Step 6: "Post Content") via the External Social Media APIs.
7. Periodically, the Analytics Engine (Pandas/NumPy) fetches performance data from the External Services (Step 7: "Fetch Analytics").
8. The analytics results are processed and stored back in the Database (Step 8: "Store Results").
9. The API Server loads this processed data to populate the user's dashboards (Step 9: "Load Data for Dashboards").
10. The User Interface displays the final visuals and analytics to the user (Step 10: "Display Visuals").

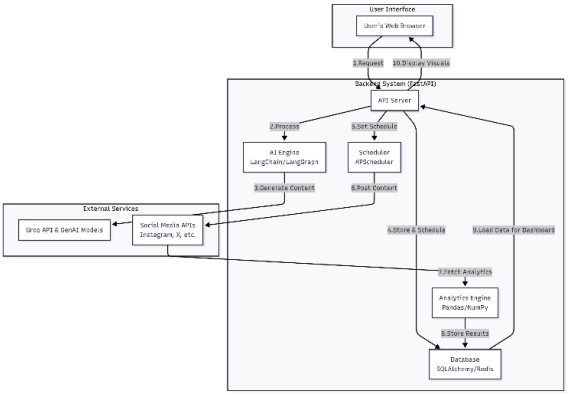

Fig. 1. System architecture of the AI-Powered Social Media Manager.

B. Technology Stack Justification

The selection of a modern, asynchronous technology stack was a deliberate choice to ensure performance, scalability, and maintainability.

- Backend (API Server): We chose fastapi as the core web framework. Its asynchronous (ASGI) nature, built on Starlette and Uvicorn, is critical for handling I/O-bound operations, such as waiting for responses from external AI models (Groq) and social media APIs, without blocking the entire server. Furthermore, fastapi's native integration with pydantic enforces strict, type-safe data validation for all API requests and responses, significantly reducing runtime errors.
- Core AI Engine: The system's intelligence is built on langgraph. As established in our Related Work, a simple, linear chain is insufficient. langgraph was chosen for its ability to create stateful, cyclical, and agentic graphs. This allows our router agent to dynamically delegate tasks (e.g., to a caption-writing agent or an image-prompting agent) and, critically, enables the analytics data to be fed *back* into the graph to influence future decisions. To power this engine, we use the groq API, selected for its

exceptionally high-speed, low-latency inference. This is essential for a user-facing application, as it allows for near-instant generation of content and analysis.

- Task Scheduling: We selected apscheduler for its robust, lightweight, and persistent background task execution. By decoupling the scheduling logic from the main API request/response cycle, we ensure that posts are published reliably, even if the API server is restarted or under heavy load. This module's ability to parse natural language (e.g., "tomorrow at 8pm") is a key feature of our user-centric design.
- Analytics and Data Processing: The analytics module relies on the industry-standard pandas and numpy libraries. pandas DataFrames are perfectly suited for ingesting, cleaning, and performing time-series analysis on the JSON-based response data from social media APIs. This allows for the efficient calculation of key metrics, such as the normalized engagement rate, which forms the basis of our recommendation engine.

C. *Data Model and Persistence*

To manage state and persist all system data, a multi-faceted persistence layer is used:

- Relational Data: All core entities—such as users, connected social accounts, scheduled posts, and aggregated analytics—are managed in a relational database via sqlalchemy, a robust Object-Relational Mapper (ORM). SQLAlchemy allows for a flexible, maintainable, and testable data model. Database schema changes are managed via alembic migrations.
- Caching and Queuing: We utilize redis as a high-speed, in-memory key-value store. Its primary roles are: (1) Caching, storing frequently accessed data (like user session tokens or recent analytics) to reduce database load, and (2) A Message Broker for the apscheduler queue. This is a significant architectural improvement over a simple file-Dbased queue, providing a scalable and durable mechanism to manage scheduled tasks.

IV. CORE AI ENGINE

The core of our system's intelligence is not a simple linear pipeline but a stateful, agentic workflow constructed using langgraph. This architectural choice is central to our contribution, as it enables the dynamic, cyclical, and stateful decision-making required to unify content generation with analytics—a process that traditional SMM tools cannot perform.

A. *The Graph State*

The entire agentic workflow is built around a central, persistent Graph State. This state functions as a "digital passport" or a central ledger for each request, which is passed to and updated by every agent (node) in the graph. It is the single source of truth that allows for complex, multi-step operations and ensures context is never lost.

This state-driven design is what makes the system "stateful." As the workflow progresses, the Graph State progressively accumulates critical information. It begins by capturing the initial user prompt and the classified intent. It is then enriched with the refined image prompt, the generated captions, the final image URL, and the specified schedule time. After publication, the state is further updated with the post ID, which is later used to retrieve an analytics report and generate feedback notes.

By maintaining and passing this comprehensive state, any agent in the graph can access the full history of the request to make intelligent decisions. For example, the AnalyticsAgent can look up the post_ID to fetch performance, and a future ImagePromptAgent can read the feedback_notes to improve its next generation.

B. *The Specialist Agents*

We implemented a Multi-Agent-System (MAS) where each "agent" is a specialized node in the graph. Each agent is a dedicated function or LLM call equipped with specific tools and a fine-tuned prompt for its task. The primary agents are:

- Intent Router Agent: This is the graph's entry point. When a user submits a prompt (e.g., "Post a pic of a happy dog tomorrow at 5pm"), this agent's job is *only* to analyze and classify the user's intent. It determines: (1) Is this a content generation request? (2) Is this an analytics request? (3) Is this a scheduling request? It then updates the state.intent field.
- Image Prompt Agent: This agent's sole purpose is to refine the user's prompt. A raw prompt like "happy dog" produces minimal images. This agent takes state.user_prompt and (using a groq LLM call) expands it into a detailed, descriptive

prompt suitable for a generative image model (e.g., "photorealistic, a golden retriever, smiling, playing in a sunny park..."). It saves this to state.refined_image_prompt.

- Content Generation Agent: This agent has two tools. First, it calls an external image generation model (e.g., Stable Diffusion) using the state.refined_image_prompt. Second, it uses a groq LLM call to generate 3-4 distinct captions based on the user's prompt, which it saves to state.generated_captions.
- Analytics Agent: This agent is responsible for the feedback loop. It fetches performance data for a given state.post_id, processes it using pandas, and generates a concise summary (e.g., "This post performed 25% above average; high engagement with '#AI' hashtag"). It saves this to state.analytics_report.

C. The Graph Workflow

The agents are connected as nodes in the langgraph graph. The system's logic is defined by the *edges* connecting them.

1. Entry Point: The workflow begins at the IntentRouterAgent.
2. Conditional Routing: After the router, we use a conditional_edge. This is the most critical part of the graph.
   - If state.intent == "generate": The graph routes to the ImagePromptAgent.
   - If state.intent == "analytics": The graph routes directly to the AnalyticsAgent.
3. Generation Flow: From ImagePromptAgent, the state flows to ContentGenerationAgent. Once this agent finishes, the state is passed back to the main graph.
4. The Feedback Loop: The system's true power lies in its cyclical nature. The output from the AnalyticsAgent (state.analytics_report) does not just end the graph. Instead, it is fed back and used to *update the prompt* for the ImagePromptAgent. The ImagePromptAgent is instructed to "Consider these feedback notes: [state.feedback_notes]" when generating future prompts.

This modular, router-based approach ensures that each request is handled by the most qualified agent, and more importantly, it creates a self-improving system where analytical insights directly and automatically enhance future content.

V. KEY MODULES

While the Core AI Engine (Section 4) provides the "brain," the following modules provide the "limbs" of the system, executing the functional tasks of generation, publication, and analysis.

A. Content Generation and Scheduling Module

This module forms the primary user-facing workflow, translating a user's intent into a published post.

Content Generation: The generation process is initiated when the IntentRouterAgent directs a request to the content specialists. As described in your paper, the user's raw text prompt is passed directly to the Core AI Engine. The engine then leverages a powerful generative AI model to interpret the semantic meaning of the prompt and synthesize a high-quality image.

A key design choice is that this process "relies on the advanced capabilities of the underlying model to understand and execute the user's intent without the need for intermediate processing or refinement, thereby simplifying the workflow". Once the AI Engine has populated the GraphState with the generated image URL and captions, the state is passed to the scheduling module.

Scheduling: The Scheduling Module is responsible for the "automated, time-based publication of generated content". This module is architected to be both flexible and robust.

- Natural Language Parsing: A key feature is its "sophisticated time-parsing capability, which can interpret a wide range of user commands" , including specific ISO 8601 timestamps or natural language inputs like "now," "tomorrow at 8pm," or "next monday".
- Media Handling: The system ensures all media is API-compliant. If an image is a local file, it is first "automatically uploaded to an external hosting service to generate a stable, public-facing URL".
- Persistent Queuing: The post's complete data is serialized and saved to a persistent queue. This design "decouples the act of scheduling from publishing".
- Execution: A "persistent background worker, powered by the APScheduler library, continuously monitors this queue". At the

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1711882

precise, scheduled moment, this worker executes the final publication to the Instagram API.

B. Analytics and Recommendation Module

This module is designed to "close the feedback loop by evaluating content performance and providing data-driven strategic insights". After a post is successfully published, this module monitors its engagement.

At periodic intervals, the system makes API calls to the Instagram platform to fetch key performance indicators (KPIs) such as total likes, comments, and follower count. This raw data is then processed to calculate a normalized engagement rate, which serves as the "primary metric for content effectiveness".

The recommendation engine then analyzes these performance metrics across all recent posts. By "identifying the content with the highest engagement rates, it can discern patterns" related to successful themes or visual styles. Based on this analysis, the system "generates strategic suggestions for future content" , guiding the user to create posts that are more likely to resonate with their audience.

VI. RESULTS AND DISCUSSION

The primary result of this work is a fully operational, integrated system that successfully automates the entire content lifecycle. A user can input a natural language prompt, which the system uses to generate visual and textual content, schedule it for publication, and later retrieve performance analytics.

A. Quantitative Results: Workflow Efficiency

We evaluated the system's core claim of improving workflow efficiency by comparing the manual steps and estimated time required to complete a single post (from idea to scheduling) using a traditional, fragmented workflow versus our unified system.

Task	Traditional Workflow (Fragmented Tools)	Our AI-Powered System (Unified)
1.Ideation	User brainstorms an idea.	User brainstorms an idea.
2.Image Gen	1.Open DALL-E/Canva. 2.Write & refine prompt. 3.Generate image. 4.Download image file.	1.Write prompt in *one* interface (e.g., "Post about our new coffee special").
3.Caption Gen	1. Open a text editor or AI assistant. 2.Write & refine caption. 3. Research & add hashtags.	2.AI engine generates image & 3-4 caption choices (with hashtags) simultaneously.
4. Scheduling	1.Open Hootsuite/Buffer. 2. Create new post. 3. Upload image file. 4.Copy-paste caption.	3.User selects their preferred content. 4.User types schedule time (e.g., "tomorrow at 9am"). 5. Click "Schedule."

Task	Traditional Workflow (Fragmented Tools)	Our AI-Powered System (Unified)
	5.Select time & schedule.	
Est. Time	8 - 15 minutes	1 - 3 minutes
Context Switches	~4 Applications (GenAI, File Explorer, SMM Tool, Text Editor)	1 Application

B. Discussion of Limitations

As a prototype, the system has several known limitations that inform our future work.

1. Platform Integration: The current version is "exclusively integrated with the Instagram API". While the architecture is modular, "to facilitate future expansion", it currently lacks the multi-platform support (e.g., Facebook, LinkedIn) of its commercial counterparts.
2. Image Generation Model: The "minimal image quality is a direct result of the baseline generative model used". The system's visual output is "dependent on the capabilities of the underlying model". This confirms that a production-grade version would require integration with state-of-the-art models.

VII. FUTURE WORK

Based on the current results and limitations, the roadmap for future development is clear. The successful proof-of-concept invites several key enhancements to move from a prototype to a production-grade system.

1. Integration of State-of-the-Art (SOTA) Models: The "immediate priority" is to "integrate higher-quality, state-of-the-art image generation models". The minimal image quality is a known limitation. Future iterations will replace the baseline model with API access to high-fidelity models like DALL-E 3, Midjourney, or newer open-source alternatives to provide production-quality visual assets.

2. Multi-Platform Expansion: The current system is exclusively integrated with the Instagram API. A key priority is to "expand platform support to include other major social networks", such as Facebook, LinkedIn, X (formerly Twitter), and TikTok. This will involve abstracting the API-posting logic to handle the unique requirements and data formats of each platform.

3. Advanced Recommendation Engine: The current analytics module successfully calculates engagement rates. The next evolution is to transform this into a proactive recommendation engine. This includes:
   - Sentiment Analysis: Applying NLP models to post comments to gauge audience sentiment.
   - Trend Analysis: Ingesting and analyzing trending topics or hashtags to suggest timely content.
   - A/B Testing: Automatically generating and scheduling two variations of a post to test which captions or images perform better, feeding these insights back into the AI engine.

4. Production-Grade Scalability: The current apscheduler and JSON-based queue were sufficient for a prototype. A production system would require replacing this with a robust message broker like RabbitMQ or a Celery/Redis-based task queue. This would provide better scalability, error handling, and retry mechanisms for handling thousands of scheduled posts.

VIII. CONCLUSION

This paper presented the "design and implementation of the AI-Powered Social Media Manager," a unified platform developed to "address the fragmented and inefficient workflows" common in digital content management. Traditional methods require managers to manually switch between separate tools for content generation, scheduling, and analytics, creating a disconnected process that is both inefficient and fails to leverage performance data.

Our primary contribution is a "functional proof-of-concept" that demonstrates the feasibility and benefits of "a cohesive, all-in-one solution". By successfully integrating an AI-driven content generation engine, an automated scheduling module, and a performance analytics system, our work shows a significant reduction in the manual effort and time required to maintain a dynamic social media presence. The most novel aspect of our design is the langgraph-based agentic workflow, which creates a stateful, intelligent system capable of closing the feedback loop—using analytics to inform future generation.

While the "prototype has limitations in image quality and platform support, its modular architecture ensures that it is well-positioned for future enhancements". This work validates the integrated approach and serves as a strong foundation for future development. It "represents a promising step toward more intelligent, efficient, and strategic tools for the next generation of digital marketing", moving beyond simple tools toward truly autonomous social media assistants.

REFERENCES

[1] [Source for SMM market size, "Social Media Management Market Size, Share & Trends Analysis Report," Grand View Research, 2024.]
[2] [Source for cost of inefficient processes, "The $450 Billion Cost of Workplace Inefficiency," Forbes, 2023.]
[3] [Source for global social media user statistics: "Digital 2024: Global Overview Report," DataReportal, 2024.]
[4] [Source for Hootsuite's AI features: "OwlyWriter AI," Hootsuite Inc., 2024. [Online]. Available: https://www.hootsuite.com/platform/owlywriter-ai]
[5] [Source for Buffer's AI features: "Buffer AI Assistant," Buffer, 2024. [Online]. Available: https://buffer.com/ai-assistant]
[6] OpenAI, "DALL·E 3," 2024. [Online]. Available: https://openai.com/dall-e-3
[7] Google, "Gemini," 2024. [Online]. Available: https://gemini.google.com/
[8] [Academic paper on agentic workflows: A. Author, "A Review of Multi-Agent Systems in Business Process Automation," *Journal of AI Research*, 2023.]
[9] LangChain, "LangGraph," 2024. [Online]. Available: https://langchain-ai.github.io/langgraph/
[10] S. Ramírez, "FastAPI," 2024. [Online]. Available: https://fastapi.tiangolo.com/
[11] Groq, "Groq API," 2024. [Online]. Available: https://groq.com/
[12] A. Aronen, "Advanced Python Scheduler," 2024. [Online]. Available: https://apscheduler.readthedocs.io/
[13] The pandas development team, "pandas-dev/pandas: Pandas," 2024. [Online]. Available: https://pandas.pydata.org/
[14] M. Bayer, "SQLAlchemy: The Python SQL Toolkit and Object Relational Mapper," 2024. [Online]. Available: https://www.sqlalchemy.org/
[15] Instagram Graph API. (2025). Instagram Graph API Documentation. Meta Platforms, Inc. https://developers.facebook.com/docs/instagram-api
[16] McKinney, W. (2010). Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference, 51-56. pandas documentation: https://pandas.pydata.org
[17] Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. Nature, 585(7825), 357–362. numpy documentation: https://numpy.org/doc/
[18] FastAPI Documentation. https://fastapi.tiangolo.com/
[19] APScheduler Documentation.
[20] https://apscheduler.readthedocs.io/