

# **AUTOMATING THE CREATION OF 700 BLOGS MONTHLY**

---

## **AI-DRIVEN APPROACH**

# PROJECT OBJECTIVE

Automate end-to-end blog generation for a FinTech company, with the generation of 700 blog posts per month using AI and Python

## GENERATING QUALITY CONTENT

The first challenge was to generate high quality textual content that is engaging and relevant.

The solution to this relies on keyword extraction and AI content generation



## ENSURING SEO COMPLIANCE

The second challenge was to ensure that the outputted content respected the current SEO best practices and was of high quality.

The solution to this relies on smart SEO and readability checks, AI generated improvements, and internal linking strategies.



## SCALING CONTENT PRODUCTION

Finally, the challenge was to ensure a structure that allows for scalability and workflow automation while keeping in mind costs and technological constraints

# WORKFLOW OVERVIEW



# TECHNICAL HIGHLIGHTS



## TOOLS & TECHNIQUES

Keyword extraction and content generation:

- ❑ OpenAI API
- ❑ Google Search Scraping (BeautifulSoup, NLTK)

SEO Optimization:

- ❑ Python (TextBlob, Textstat)
- ❑ NLP models

Internal Linking:

- ❑ TF-IDF
- ❑ Cosine Similarity (scikit-learn)

CMS integration:

- ❑ Requests API
- ❑ CMS REST APIs



## KEY FEATURES

- ❑ Parallel processing for efficiency
- ❑ Environment variables management
- ❑ Multiple publishing and save options
- ❑ AI model selection for cost control
- ❑ Modular design for scalability and extensibility:

```
project_folder/
|-- README.md           # Instructions
|-- requirements.txt     # Required python libraries
|-- main.py             # Main script to execute the automation pipeline
|-- keyword_extraction.py # Extract keywords using AI and web scraping
|-- oai_content_generation.py # AI-based blog content generation
|-- seo_optimization.py  # SEO analysis and fixes
|-- internal_linking.py  # Internal linking strategy
|-- cms_integration.py   # Publishing to CMS
|-- helpers.py          # Utility functions for saving outputs
|-- .env                # Environment variables for credentials
|-- .gitignore           # Instructions for git
|-- outputs/            # Generated blog content
|-- data/               # Folder for previously created blog posts
|-- utils/
|   └── helpers.py      # Helper functions
```

---

# ROADMAP FOR ML INTEGRATION

While the project uses AI and NLP, several other ML techniques could enhance the process in the future

## RULE-BASED SEO AND AI CONTENT GENERATION



Foundational system now in place

## IMPLEMENT CONTENT QUALITY CLASSIFIERS



Improved content quality by utilizing ML model (logistic regression, random forest classifier, XGBoost) to classify low-quality content

## ADD PERSONALIZATION AND TREND ANALYSIS



More relevant content for users with clustering algorithms (K-Means, LDA, Bayesian Non-Parametrics) to cluster users and their interests, and personalize the suggestion of blog posts

## DEVELOP ADAPTIVE LEARNING MODELS



Incorporate user engagement data to refine content generation models over time. Construct ML algorithms to predict topics that will generate good engagement in the future

00

01

02

03

---

# SUMMING UP

To sum up, here are the results of the project and the next steps

## RESULTS



### End-to-end blog automation

- ❑ Automated pipeline capable of generating 700 blogs/month.
- ❑ Cost-efficient processing through batch execution and parallelization.

## SCALABILITY



### Code Structure that supports improvements

- ❑ Supports implementation of additional CMS platforms with minimal effort.
- ❑ Modular codebase allows future ML integration.

## NEXT STEPS



### CMS, ML, PIPELINE automation

- ❑ Expand CMS compatibility implement ML.
- ❑ Real-time dashboards for performance monitoring
- ❑ Fully automate the whole pipeline with cloud, airflow, AWS

# CONCLUSIONS & TAKEAWAYS

The present document provides a high-level description of the project and the code for blog post automation.

## The outcome of the project is:

- ✓ fully automated blog generation tailored for FinTech
- ✓ scalable, cost-conscious, and with modular implementation
- ✓ ready for future enhancements  
(e.g., dashboards, adaptive ML models)



*For more in-depth information, a word document describing the project and the code are available*