COMP4801

Final Year Project

Universal GenAI Agent – Integrated Interactive Intelligence

Interim Report

Group: FYP24100

The University of Hong Kong

Department of Computer Science

Supervised By:

Prof. Yu Tao

Student:

Fung Ho Sang (3035927386)

Date of Submission: 26th January 2025

Abstract

This project addresses the pervasive issue of information overload, a common challenge in today's digital world where individuals and organizations struggle to obtain and process large amounts of data every day. The Universal GenAI Agent aims to alleviate this problem by developing an intelligent content delivery system that automates content retrieval from selected sources and uses Google's Gemini GenAI for summarization. By providing personalized, concise updates in real time, the system reduces the time and effort required to process information while enhancing user efficiency and decision-making. The Universal GenAI Agent empowers users to stay informed by providing timely, summarized updates based on their specific preferences and interests. This is particularly beneficial in real-world scenarios such as job searching, where users can receive notifications about job openings and company updates, or in industry news monitoring scenarios, where timely and processed insights can drive informed decisions. Preliminary results suggest that the Universal GenAI Agent can integrate multiple services. By automating content extraction and summarization, the system significantly reduces manual processes required in information retrieval, saves users valuable time and delivers actionable insights. Its scalability and adaptability make it suitable for a wide range of applications beyond job seeking, including tracking travel alerts, monitoring industry updates, or staying updated on personal interests. By addressing the inefficiencies of traditional information retrieval processes, this project offers a practical and innovative solution to information overload. It not only enhances the user experience but also highlights the potential of AI-driven personalization in delivering smarter, more efficient content solutions.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	**Full Form**
API	Application Programming Interface
BSON	Binary JavaScript Object Notation
CAPTCHAs	Completely Automated Public Turing Test to Tell Computers and Humans Apart
GenAI	Generative Artificial Intelligence
LLM	Large Language Model
JSON	JavaScript Object Notation
RSS	RDF Site Summary

1. Introduction

Information overload is a pervasive issue in today's digital age, affecting both individuals and organizations. Information overload refers to the overwhelming amount of data that negatively affects decision-making, productivity, and mental well-being [1]. Research by OpenText highlights the scale of this issue: 80% of respondents report experiencing information overload, with 27% needing to access more than 11 tools daily—up from 15% two years ago. Nearly half spend over an hour daily searching for data, while 42% find critical information scattered across sources. [2]. An evident example of this challenge is the job-seeking process. Job seekers often maintain a targeted list of companies but face significant difficulty monitoring career pages and tracking relevant news across multiple platforms. This fragmented flow of information not only leads to decision fatigue and reduces the efficiency of their job search efforts, but also very time consuming. Moreover, the lack of standardized tools, such as RSS (RDF Site Summary) feeds, hinders automated information retrieval. Users must manually monitor updates, increasing the risk of missing critical information.

With the recent development in GenAI (Generative Artificial Intelligence) tools that utilize Large Language Models (LLM) to understand texts and output responses, it may also be integrated in a content delivery system to provide analyzed information. This Universal GenAI Agent project aims to address the challenges of information overload by developing a personalized content delivery system that automates content retrieval and uses GenAI to summarize content into a concise, digested form. The motivation for this project stems from the need to alleviate the cognitive burden of searching through vast amounts of information. This project proposes an intelligent, automated content delivery agent powered by GenAI. By integrating advanced content retrieval, analysis, and personalization, the system can transform the vast amounts of information into highly relevant, context-aware, and user-specific outputs.

There are three main objectives to this project:

1) To develop an all-Purpose, GenAI-embedded agent that integrates seamlessly with a wide range of third-party services, delivers tailored content to users regularly and allows easy user interaction to efficiently perform routine actions.
2) To enhance user engagement and satisfaction by providing personalized information.
3) To explore new use cases of GenAI in content delivery that can improve efficiency.

The main deliverable for this project is the development of the Universal GenAI Agent, a comprehensive software system designed to address information overload through intelligent content delivery. The system integrates five key components: a web scraping module capable of real-time data extraction, a GenAI service connector for personalized content summarization, a database for storing user-specific preferences, a scalable notification system to deliver updates to multiple users efficiently, and a Discord AI Chatbot agent to provide users with additional information upon request. Completed course deliverables include a detailed project plan, a project web page with regular progress updates, the first FYP presentation, and an interim report. Upcoming course deliverables include a final report and a final presentation.

Similar research shows significant improvements in user engagement through AI-driven personalization. Study by Sodiya et al. [3] illustrates that AI-driven personalization can boost user engagement by presenting relevant information based on user behavior, significantly improving overall satisfaction. Research by Smith et al. [4] indicates that AI systems can analyze user behaviors to provide personalized recommendations, saving time and minimizing the effort needed to retrieve relevant content. These studies suggest potential advantages of implementing a personalized content delivery system. Despite these promising findings, there remains a gap in the practical application of AI-driven personalization. The current studies primarily focused on e-commerce and entertainment rather than real-time information retrieval, and very few studies specifically explore dynamic contexts such as job hunting or industry news monitoring, where users need constantly updated, summarized content. Existing systems also tend to be static, lacking the flexibility to adapt to users' evolving needs. The Universal GenAI Agent addresses this gap by integrating real-time web scraping and GenAI-powered summarization to deliver concise, personalized content in real time, reducing information overload and enhancing user efficiency.

In the remaining sections, Section 2 outlines the project methodology, with details of the technologies and design decisions involved in the development of the Universal GenAI Agent. Section 3 highlights preliminary results, showcasing the system’s potential to address information overload with real-time content retrieval and summarization. Section 4 presents the project schedule and timeline, summarizing completed tasks and future milestones. Finally, Section 5 provides the conclusion, summarizing the key contributions of the project and its potential implications for addressing information overload through AI-driven solutions.

2. Methodology

In Section 2, there are three subsections: Section 2.1 gives the overall design of system architecture and illustrates the logic flow between different components in the Universal GenAI Agent. Section 2.2 justifies the specific design choices and decisions made to each component. Section 2.3 illustrates the actions that will be taken to adhere to ethical standards.

2.1 System Architecture Design

The Universal GenAI Agent adopts a modular architecture, designed to support efficient development and precise debugging. The system comprises six distinct components, each performing a specific function, listed sequentially according to their roles within the overall workflow below:

1) Discord Chat Interface: Serves as the user interface, allowing users to request information, interact with the system, and receive responses in real time.
2) Cron Job Notification System: Regularly executes the workflow to push summarized content to users at specified intervals, ensuring timely updates through the Discord Chat Interface.
3) Master Server: Acts as the central coordinator, managing communication and operations between various components to ensure smooth workflow execution.
4) Selenium Web Scraping Module: Automates the retrieval of content from selected websites, including dynamic content rendered through JavaScript, to ensure comprehensive data collection.
5) NeDB Database: Stores user preferences, such as topics of interest and notification settings, as well as previously analyzed content, enabling personalized content delivery.
6) Gemini GenAI Service: Processes the retrieved content and generates concise, user-relevant summaries based on user preferences.

Figure 1 below shows the workflow between different components when the system handles a user input.

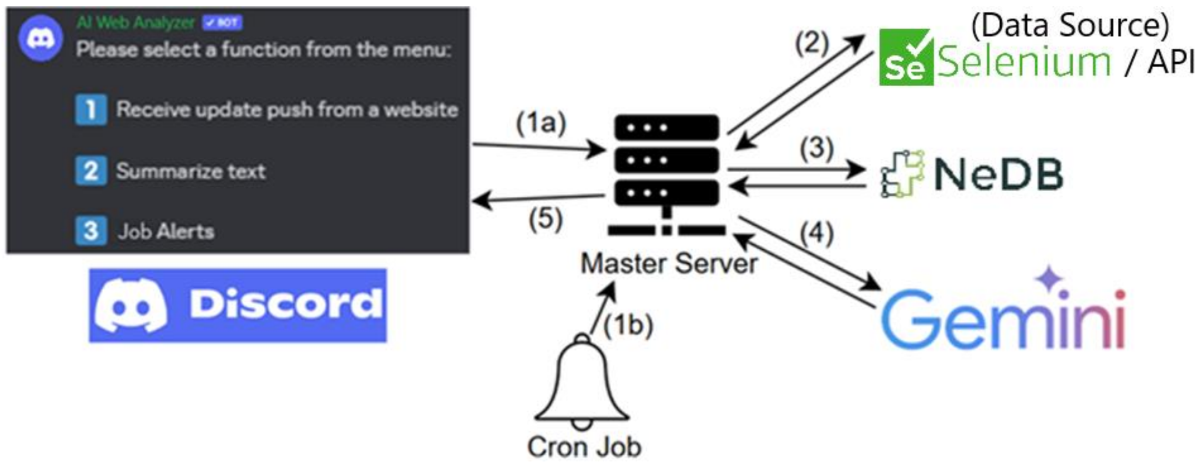


Figure 1: Workflow sequence illustrated by arrows and numbered steps. The process either begins with user input in the Discord Chat Interface (1a) or an automated cron job that regularly notifies the user (1b), then the master server will handle the request and proceed through the following steps: (2) Extract raw data from either website content extracted using Selenium or third-party data from API service providers, (3) Retrieve user detail from NeDB database, (4) Analyze and personalize content using Gemini GenAI, and (5) Output digested content to the user in the Discord Chat Interface.

The workflow of the Universal GenAI Agent (Figure 1) begins with either user input through the Discord Chat Interface (Step 1a) or an automated process initiated by the cron job notification system (Step 1b), which delivers updates based on predefined schedules or user preferences. The Master Server handles the requests and manages communication between the components. The process proceeds by obtaining the raw content to be analyzed, from either user selected websites scraped using Selenium in real time, or third-party data from API service providers (Step 2). These raw contents are then combined with the user's preferences retrieved from the NeDB database, including topics of interest, language settings, and recent queries (Step 3) to personalize the contents with individual requirements in subsequent operations. The retrieved content, alongside user preferences, is processed by the Gemini GenAI service (Step 4), which analyzes the data, generates concise and relevant summaries, and personalizes the output to align with the user's specific needs. The final summarized content is delivered to the user through the Discord Bot (Step 5), notifying the users with the newest information. Users may further engage with the system via the Discord Chat interface to ask follow-up questions (the workflow starts again with Step 1a), to retrieve additional details. This workflow is designed to provide an efficient and highly personalized content delivery system while addressing the challenges of information overload.

2.2 Engineering choices, decisions and justifications

This section illustrates and explains the choice of the six core components that the Universal GenAI Agent is built on. These components were chosen based on their technical capabilities, ease of development, and alignment with the overall goals of the system. Each subsection below would discuss the choice of each component.

2.2.1 Discord Chat Interface

The Discord Chat Interface serves as the primary user interface for the Universal GenAI Agent, enabling users to interact with the system in real time. Discord was chosen due to its popularity as an instant messaging software. Compared to traditional web-based interfaces, Discord provides a familiar platform for users, reducing onboarding time and simplifying communication in the format of daily messaging. This interface allows users to submit queries, receive personalized content, and interact with the AI agent for follow-up questions, creating a dynamic and engaging user experience. Its real-time messaging capabilities also ensure that users can access updates and responses quickly and efficiently. Discord provided a robust Application Programming Interface (API) with clear documentation of the data models [5], allowing the Discord AI agent to interact with the master server.

2.2.2 Cron Job Notification System

The Cron Job Notification System will be implemented to deliver summarized content to users at specified intervals. Cron jobs enable automated, time-based task scheduling. The system will execute the workflow at predefined intervals to continuously monitor websites for content updates. When new or relevant content is detected by the web scraping module, notifications will be triggered in real time, ensuring that users are notified promptly without requiring manual intervention.

2.2.3 Master Server

The Master Server acts as the central coordinator, managing communication and operations among the various components of the system. This centralized architecture was selected to ensure smooth workflow execution and to simplify the integration of different modules. The Master Server is responsible for directing data flow, initiating processes such as triggering web scraping or content summarization, and maintaining system reliability via detailed logging and robust error case handling. By centralizing control, the server minimizes latency and ensures that all components work cohesively. Additionally, the modular nature of the server design facilitates scalability, allowing for future expansion or integration of new features without disrupting the existing workflow.

The Master Server has been implemented using Node.js v.20.15.0, a JavaScript runtime widely used in the transfer of JSON (JavaScript Object Notation), the payload format used in communication between different components, to allow smooth data flow. Node.js was also chosen for its asynchronous, event-driven architecture, that efficiently handles multiple concurrent tasks [6]. The system requires parallel processing to process requests from multiple concurrent users and interaction with other components, such as web scraping, API calls, and database. The native support to asynchronous programming and non-blocking execution of Node.js is crucial for the Master Server to handle multiple tasks simultaneously.

2.2.4 Web Scraping and Content Extraction

Web scraping is a vital component of the Universal GenAI Agent, facilitating the automated extraction of relevant content from various websites. The web scraping algorithm is based on Selenium, as it can fetch dynamic content generated by JavaScript, simulating user interactions to access information that is not available in static HTML. This dynamic scraping process can effectively gather a wide range of content types, including job postings and articles, fulfilling existing requirements and allowing for future expansion of features that require complex content retrieval.

2.2.5 NeDB Database

The database is designed to store user-specific information essential for personalizing the GenAI output for every user. This includes user IDs, preferences such as topics of interest, notification frequency, language settings, and recent queries. By capturing these details, the system can deliver tailored content that aligns with the user’s interests and preferred communication methods. The database structure is also designed to accommodate additional user-specific data in the future, ensuring flexibility as personalization needs evolve.

The database will be implemented using NeDB due to its lightweight, schema-less structure and compatibility with JSON data format that enables seamless data mapping with the JSON payload of API communication between system components, simplifying data storage and retrieval.

2.2.6 GenAI Integration for Summarization

The integration of GenAI utilizes Google's Gemini API, specifically the Gemini 1.5 Flash API [8]. The choice is a cloud-based GenAI instead of a local LLM. Locally run models still require significant computational resources such as high-end GPUs, especially for large models. Additionally, maintaining and updating the model, managing dependencies, and ensuring scalability can be challenging. Cloud-based models offer regular updates and eliminate hardware performance limitation, making them more practical for most use cases compared to running local models.

Cloud-based models may have different usage limits. Table 1 below shows the comparison in the rate limit and usage limit of the free tier between the API of OpenAI's ChatGPT-3.5-Turbo [7] and Google's Gemini 1.5 Flash [8].

Table 1: Comparison between OpenAI's ChatGPT API and Google's Gemini API [7-8]

	ChatGPT-3.5-Turbo API	Gemini 1.5 Flash API
Rate Limit	3 Requests per minute 200 Requests per day	15 Requests per minute 1,500 Requests per day
Usage Limit	40,000 Tokens per minute	1,000,000 Tokens per minute
Pricing	Free of charge	Free of charge

Referring to Table 1, Gemini has a much higher usage limit than that of ChatGPT, offering 1,000,000 tokens per minute, which is 25 times greater than ChatGPT's 40,000 tokens per minute. This is the major reason for choosing Gemini. A larger number of tokens indicates longer text can be input to the model. As websites usually have very long texts, it is crucial that the GenAI model supports a long length of content. In this case, Gemini is vastly superior to ChatGPT. Additionally, Gemini supports a higher rate limit of 15 requests per minute and 1,500 requests per day, compared to ChatGPT's 3 requests per minute and 200 requests per day. This higher rate limit ensures the system can handle multiple simultaneous user requests efficiently, reducing the risk of delays or bottlenecks. Combined with its superior token capacity, Gemini enables the Universal GenAI Agent to deliver consistent and reliable performance, even under heavy usage.

The Gemini connector has been developed using Node.js since communication with Gemini's API server uses the JSON data structure. This choice also ensures consistency with other system components, simplifying integration and maintaining a unified development framework.

2.3 Challenges and Mitigations

While the project has progressed smoothly so far, certain challenges are anticipated as development continues. These potential difficulties, along with their proposed mitigation strategies, are outlined below.

2.3.1 Web Scraping Challenges

Dynamic websites often implement anti-scraping measures such as CAPTCHAs (Completely Automated Public Turing Test to Tell Computers and Humans Apart), rate limits, or JavaScript-based content rendering, which may hinder the effectiveness of the scraping module. To address this, the web scraping module will utilize Selenium to simulate user interactions and retrieve dynamic content. Additionally, rotating proxies and request delays will be implemented to avoid triggering anti-scraping mechanisms. The system will also adhere to website terms of service and only scrape publicly available content to ensure ethical compliance.

2.3.2 System Reliability

There may be unexpected errors in modules such as web scraping, API communication, or the cron job system that could disrupt the overall workflow. To mitigate this, the Master Server will include robust error-handling mechanisms and detailed logging to identify the point of failure and resolve issues promptly. Retry mechanisms will be implemented to ensure that temporary failures, such as a missed API response or a network issue, do not disrupt the overall workflow. The retry mechanisms will automatically attempt to reprocess failed tasks after a short delay, minimizing manual intervention for temporary issues. Regular monitoring and testing will help identify potential issues before they impact live operations, ensuring the system remains reliable.

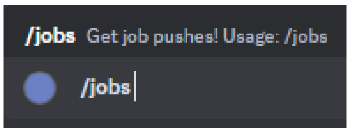
3. Preliminary Result

Preliminary results from the Universal GenAI Agent project demonstrate its potential to significantly reduce the time and effort required for users to access and digest personalized information. The Universal GenAI Agent addresses these inefficiencies by automating the retrieval and summarization of relevant content. Completed modules include the Discord Chat Interface, Selenium Web Scraping Module, and the Gemini GenAI Connector, while the development of the Cron Job Notification System and the NeDB Database is still in progress.

Implemented Discord Commands:

- /jobs: Retrieves new job postings from LinkedIn API

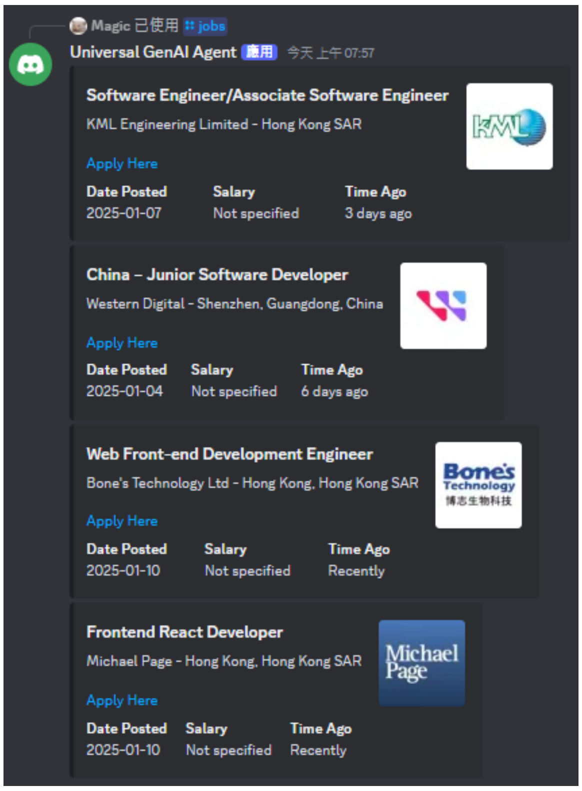
The following figures illustrate the use of the /jobs function. Figure 2 demonstrates how to execute the command by simply typing /jobs to retrieve relevant job postings from the LinkedIn API. Figure 3 displays the response received after executing this command, showcasing the available job opportunities.



/jobs Get job pushes! Usage: /jobs

/jobs

Figure 2: Example usage of the /jobs function, by typing /jobs to retrieve relevant job postings



Magic 已使用 jobs
Universal GenAI Agent 應用 今天 上午 07:57

Software Engineer/Associate Software Engineer
KML Engineering Limited - Hong Kong SAR
Apply Here
Date Posted 2025-01-07 | **Salary** Not specified | **Time Ago** 3 days ago

China – Junior Software Developer
Western Digital - Shenzhen, Guangdong, China
Apply Here
Date Posted 2025-01-04 | **Salary** Not specified | **Time Ago** 6 days ago

Web Front-end Development Engineer
Bone's Technology Ltd - Hong Kong, Hong Kong SAR
Apply Here
Date Posted 2025-01-10 | **Salary** Not specified | **Time Ago** Recently

Frontend React Developer
Michael Page - Hong Kong, Hong Kong SAR
Apply Here
Date Posted 2025-01-10 | **Salary** Not specified | **Time Ago** Recently

Figure 3: Example response after the command /jobs has been executed

- /analyze: Analyzes content from any website and answers user's follow-up questions.

The following figures demonstrate the functionality of the /analyze command. Figure 4 shows how to execute the command using the URL "scmp.com", which is a news website. Figure 5 presents the response generated from this analysis, while Figure 6 demonstrates the interaction potential and illustrates the response to the user's selected follow-up question.

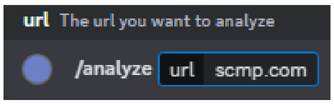
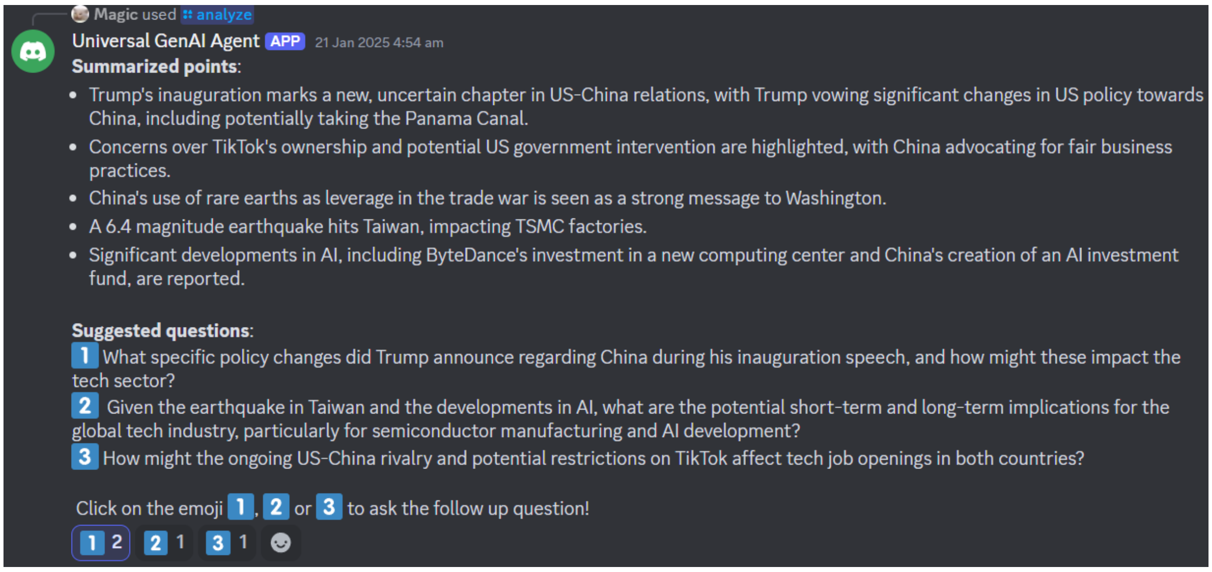


Figure 4: Example usage of the /analyze function, by typing /analyze [url] and using "scmp.com" in the url field as an example website to scrape content



Magic used analyze

Universal GenAI Agent APP 21 Jan 2025 4:54 am

Summarized points:

- Trump's inauguration marks a new, uncertain chapter in US-China relations, with Trump vowing significant changes in US policy towards China, including potentially taking the Panama Canal.
- Concerns over TikTok's ownership and potential US government intervention are highlighted, with China advocating for fair business practices.
- China's use of rare earths as leverage in the trade war is seen as a strong message to Washington.
- A 6.4 magnitude earthquake hits Taiwan, impacting TSMC factories.
- Significant developments in AI, including ByteDance's investment in a new computing center and China's creation of an AI investment fund, are reported.

Suggested questions:

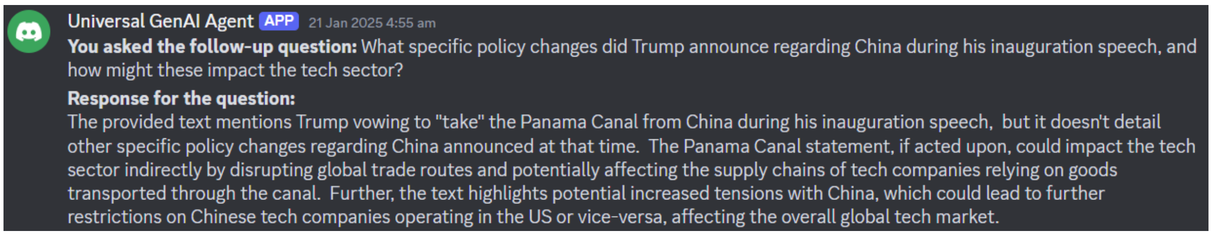
1 What specific policy changes did Trump announce regarding China during his inauguration speech, and how might these impact the tech sector?

2 Given the earthquake in Taiwan and the developments in AI, what are the potential short-term and long-term implications for the global tech industry, particularly for semiconductor manufacturing and AI development?

3 How might the ongoing US-China rivalry and potential restrictions on TikTok affect tech job openings in both countries?

Click on the emoji 1, 2 or 3 to ask the follow up question!

Figure 5: Example response after the command /analyze has been executed with url "scmp.com"



Universal GenAI Agent APP 21 Jan 2025 4:55 am

You asked the follow-up question: What specific policy changes did Trump announce regarding China during his inauguration speech, and how might these impact the tech sector?

Response for the question:

The provided text mentions Trump vowing to "take" the Panama Canal from China during his inauguration speech, but it doesn't detail other specific policy changes regarding China announced at that time. The Panama Canal statement, if acted upon, could impact the tech sector indirectly by disrupting global trade routes and potentially affecting the supply chains of tech companies relying on goods transported through the canal. Further, the text highlights potential increased tensions with China, which could lead to further restrictions on Chinese tech companies operating in the US or vice-versa, affecting the overall global tech market.

Figure 6: Example response after asking a follow-up question to the previous response of /analyze

The system's Selenium web scraping module, integrated with the Gemini GenAI service, has shown promising performance. Initial tests on 50 websites indicate that content retrieval completes in under 10 seconds, with the GenAI service generating concise summaries in the same timeframe. Once the database has been implemented, the full workflow is expected to deliver content to users in under 30 seconds. Take job seeking as an example, the system allows timely access to the latest job openings and easy-to-read descriptions, effectively reducing information overload and cognitive effort.

4. Project Schedule and Timeline

The Universal GenAI Agent project is progressing according to the planned timeline, with several key components already completed and others in progress or scheduled for development:

- Completed Tasks (August – December 2024):
    - System Modules:
        - Selenium web scraping module
        - Gemini GenAI service connector
        - Discord API gateway
    - Discord commands:
        - /analyze, /jobs

- In Progress (January – February 2025):
    - System Modules:
        - Database setup to store user preferences and support personalized delivery.
        - Cron job implementation to automate the workflow and push notifications
    - Planned Discord commands (Personal interest tracking from third-party APIs):
        - /sports: Provide live updates or scores for ongoing games in selected sports (e.g., football, basketball).
        - /flights: Check for cheap flight deals and travel alerts.
        - /cat: Get daily picture of cats (can even choose the breed).
        - /music: Fetch recommendations for songs or playlists based on user mood.
        - /events: List upcoming local events or concerts based on user interests.

- Upcoming Tasks (March – April 2025):
    - Release of the preliminary implementation for user feedback and seek improvements.
    - Testing and fine-tuning of all system components to ensure reliability, scalability, and performance under various use cases.

The project is currently on track, with all components progressing as expected. Next steps are to finish remaining tasks, conduct performance testing, and analyze results to refine the system.

5. Conclusion

The Universal GenAI Agent project offers an innovative solution to the challenge of information overload. By integrating real-time web scraping, advanced Generative AI (GenAI), and personalized content delivery, the system enables users to access meaningful, actionable insights in a fraction of the time required for manual data retrieval and processing. This innovation promotes convenience, addressing cognitive fatigue and inefficiencies in information management, and fundamentally redefines how users engage with the digital world.

The system's modular architecture — comprising the Discord Chat Interface, Selenium Web Scraper, NeDB, and Gemini GenAI — allows high adaptability in fine tuning to support diverse use cases. From job seekers seeking timely updates to individuals tracking personal interests, the Universal GenAI Agent delivers personalized, real-time content tailored to unique user needs. Preliminary results demonstrate substantial reductions in data processing time and cognitive effort, highlighting its potential to enhance user productivity and decision-making.

Ultimately, the Universal GenAI Agent represents an attempt in the application of GenAI for personalized content delivery. By addressing critical gaps in existing information retrieval systems, this project generates personalized and contextually relevant outputs that dynamically adapt to evolving user preferences. Its potential applications extend well beyond the realms of job searching and personal interests monitoring; they encompass diverse areas such as social media trend analysis, health monitoring alerts, and personalized educational content delivery.

In conclusion, the Universal GenAI Agent demonstrates the potential of GenAI technologies in optimizing workflows and enhancing decision-making across various sectors. It addresses the challenges of information overload, investigating AI's ability in creating a more efficient and personalized digital experience for users. This foundational work not only paves the way for future innovations in AI-driven content delivery systems, but also highlights the crucial role of AI in reshaping how we manage, interact with, and derive value from information.

References

[1] J. Spira, "Information Overload: A Systematic Review of the Literature," Information Overload Research Group, 2019. https://www.researchgate.net/publication/265906917_Information_Overload_A_Systematic_Literature_Review (accessed Oct. 14, 2024).

[2] S. Ono, “Unlock the information advantage to combat ‘information overload’,” OpenText Blogs, 2022. https://blogs.opentext.com/unlock-the-information-advantage-to-combat-information-overload/ (accessed Nov. 30, 2024).

[3] E. O. Sodiya, O. O. Amoo, U. J. Umoga, and A. Atadoga, "AI-driven personalization in web content delivery: A comparative study of user engagement in the USA and the UK," World Journal of Advanced Research and Reviews, vol. 21, no. 2, pp. 887–902, 2024. https://doi.org/10.30574/wjarr.2024.21.2.0502 (accessed Oct. 14, 2024).

[4] B. Smith, A. Johnson, and R. Lee, "Exploring the Benefits of AI for Content Retrieval," Journal of Information Science, vol. 45, no. 1, pp. 1-10, 2023. https://www.researchgate.net/publication/378575962_Exploring_the_Benefits_of_AI_for_Content_Retrieval (accessed Oct. 14, 2024).

[5] “API docs for bots and developers,” Discord Developer Portal, https://discord.com/developers/docs/interactions/receiving-and-responding#receiving-an-interaction (accessed Nov. 30, 2024).

[6] “Node.js - about node.js,” Node.js, https://nodejs.org/en/about (accessed Nov. 30, 2024).

[7] “OpenAI Platform,” OpenAI, 2024. https://platform.openai.com/docs/guides/rate-limits/usage-tiers?context=tier-free (accessed Oct. 16, 2024).

[8] “Gemini API Pricing | Google AI for Developers,” Google AI for Developers, 2024. https://ai.google.dev/pricing#1_5flash (accessed Oct. 16, 2024).