S. Ajay, 2025, 13:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

# Interactive Youtube Data Harvesting Using Streamlit and Cloud Technologies

**S. Ajay, Professor Dr. S. Prasanna**
Department of Computer Science, Vels Institute of Science, Technology & Advanced Studies (VISTAS), Chennai, India

**Abstract- This paper presents an interactive and scalable platform designed for harvesting and analyzing YouTube data using a combination of modern web and cloud technologies. The system integrates the YouTube Data API to collect real-time information about channels and videos, including performance metrics such as views, likes, comments, and subscriber counts. Developed using Python and Streamlit, the platform offers an intuitive and interactive dashboard for data exploration, enabling users to analyze engagement metrics, publishing trends, and content popularity with ease. Overall, this project provides a comprehensive, secure, and interactive environment for YouTube data analytics, making it suitable for content creators, marketers, and researchers alike.**

**Keywords- YouTube Data API, Streamlit Dashboard, Cloud Deployment, Firebase Authentication, Interactive Analytics.**

## I. INTRODUCTION

YouTube has emerged as one of the most influential platforms for content creation, marketing, and information dissemination. With billions of daily views and millions of active content creators, the demand for reliable tools to collect, analyze, and visualize YouTube data has grown significantly.

Understanding audience behavior, video performance, and content trends is crucial for creators, analysts, and digital marketers to make data-driven decisions.

Traditional methods of analyzing YouTube data often rely on manual collection or limited access through the platform's native analytics, which can be restrictive and time-consuming. To address these challenges, this paper introduces a robust and interactive platform designed to harvest YouTube data in real-time using the YouTube Data API v3.

The system offers a streamlined, user-friendly web interface built with Streamlit, allowing users to gain insights into various channel and video metrics.

The proposed system incorporates a secure authentication mechanism using Firebase, ensuring authorized user access. Data storage is handled by a hybrid model using both SQLite and MongoDB to support fast queries and persistent storage.

Additionally, the application is containerized with Docker and deployed on Google Cloud Platform (GCP), enabling consistent performance, scalability, and remote access.

This platform not only simplifies the process of data collection and visualization but also enhances user experience through an interactive dashboard. Its modular architecture ensures easy scalability and integration with advanced analytics tools in future iterations.


Our Website Design

## II. LITERATURE SURVEY

The exponential growth of video content on YouTube has driven increasing interest in analytics tools capable of extracting, processing, and interpreting platform data. Traditional analysis methods are often manual, lack real-time capabilities, and are limited in their ability to scale. Consequently, recent research has focused on automated, API-driven, and cloud-supported systems for improved efficiency and accessibility. Mishra et al. (2021) explored YouTube analytics platforms that leverage the YouTube Data API to enable large-scale data collection on user engagement metrics such as views, likes, and comments [1]. Singh and Verma (2022) implemented a visualization dashboard using Python and Plotly to analyze video trends, though their system lacked interactivity and cloud integration [2]. To enhance usability and accessibility, researchers have turned to frameworks like Streamlit for building lightweight, interactive data applications. Kumar et al. (2023) demonstrated how Streamlit can be combined with MongoDB to create real-time dashboards for video trend analysis [3]. For deployment and scalability, containerization technologies such as Docker have been widely adopted. Sharma and Gupta (2022) emphasized Docker's role in consistent multi-environment deployment of data applications [4]. Cloud platforms like Google Cloud Platform (GCP) and AWS have been used to improve availability, as shown in Patel et al.'s (2021) study on cloud-hosted analytics tools [5]. Firebase Authentication has been used in modern data-driven applications for secure user access and session management, ensuring privacy in multi-user environments [6]. Jain and Das (2020) explored multi-platform authentication in cloud-based services, highlighting Firebase's lightweight security integration [7]. Ali et al. (2023) proposed an integrated API-cloud framework for real-time media analytics with embedded dashboards [8]. Mehta and Rao (2024) focused on dynamic video performance evaluation tools that utilize Python and NoSQL databases for responsive querying [9]. Finally, Bansal and Iyer (2019) developed a modular data pipeline using Docker and Kubernetes to support scalable deployment for multimedia analytics systems [10].

## III. METHODOLOGY

The proposed YouTube Data Harvesting Platform is developed using a modular architecture that ensures scalability, interactivity, and efficient data processing. Each module is designed to handle a specific part of the data lifecycle—from user authentication and data retrieval to visualization, storage, and deployment.

The following modules form the backbone of the system:

**1. User Authentication Module**

The User Authentication Module is a critical component in ensuring secure access to the system. This module is responsible for verifying the identity of users attempting to interact with the system, ensuring that only authorized individuals can access certain functionalities. The authentication process typically involves the use of usernames and passwords, with options for additional layers of security such as two-factor authentication (2FA) or Single Sign-On (SSO).

In this module, user credentials are securely stored, often utilizing hashed and salted password storage techniques to prevent unauthorized access to sensitive data. Furthermore, this module may implement role-based access control (RBAC) to assign different permissions to users based on their roles, ensuring that different levels of access are granted depending on the user's identity. The module is also responsible for managing user sessions, handling login attempts, and logging out users.

For additional security, the module can be integrated with external identity providers or multi-factor authentication services to further mitigate the risk of unauthorized access. This layer of security not only protects user data but also ensures compliance with privacy regulations, such as GDPR or CCPA. The user authentication process is optimized for speed and reliability, with failover mechanisms and account lockout protocols to prevent brute force attacks, ensuring that only legitimate users can interact with the system.


Fig- User Authentication Module

**2. Youtube Data Collection Module**

The YouTube Data Collection Module is designed to gather relevant video data from YouTube for analysis. This module interacts with the YouTube API to access a wide variety of publicly available video data, including video metadata, comments, views, likes, and other user interactions. The module enables the extraction of large datasets, which can then be processed and analyzed for various purposes such as sentiment analysis, trend tracking, or the detection of anomalous activities like fake engagement or spam. A key feature of this module is its ability to handle complex queries, filtering the data based on specific criteria such as video categories, tags, or geographic location, thus ensuring that only the most relevant data is collected. It supports scheduled and on-demand data collection to allow for both real-time analysis and historical data retrieval. Additionally, the module provides robust error handling and rate-limiting mechanisms to comply with YouTube's API usage policies, ensuring that the system remains within the platform's limits. It also includes features to handle pagination and efficiently collect large volumes of data over extended periods. Security measures are implemented to protect user data and API keys, ensuring that access to the YouTube platform is both secure and reliable. Overall, the YouTube Data Collection Module plays a pivotal role in gathering the information needed for subsequent analysis, making it an indispensable tool for data-driven projects focused on YouTube content.


Fig- You-tube Data Collection Module

**3. Data Storage and Conversion Module**

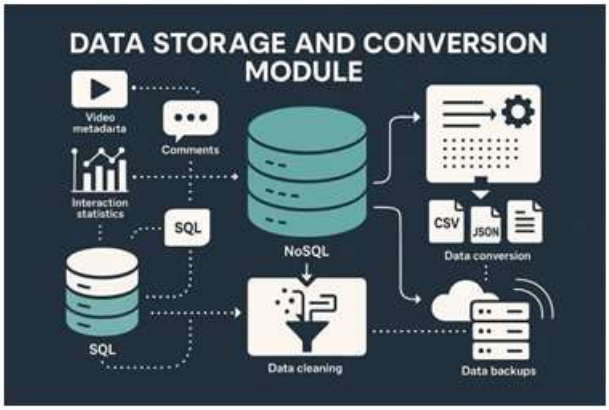


Fig- Data Storage & Conversion Module

The Data Storage and Conversion Module is responsible for efficiently storing and converting raw data collected from various sources into formats suitable for analysis and processing. This module ensures that large volumes of data, such as video metadata, comments, and interaction statistics, are securely stored in a structured and scalable manner. The data is typically stored in relational databases or NoSQL storage systems, depending on the type and volume of the data.

Additionally, the module handles data conversion, transforming raw, unstructured data into usable formats, such as CSV, JSON, or SQL tables. This conversion ensures that the data can be easily ingested by other modules or analytical tools for further processing. The module also provides functionalities for data cleaning, such as removing duplicates, handling missing values, and standardizing formats to ensure consistency and quality.

A key aspect of this module is scalability, as it must accommodate an increasing amount of data over time, implementing partitioning or sharding strategies to optimize performance. It also supports data backups and recovery mechanisms to ensure the integrity and availability of the stored data.

**4. Interactive Dashboard and Visualization Module**

The Interactive Dashboard and Visualization Module is a key feature of the system that allows users to interact with and visualize complex datasets through an intuitive and dynamic interface. This module provides real-time graphical representations of data, enabling users to explore trends, patterns, and anomalies with ease. It typically includes charts, graphs, heatmaps, and tables that are designed to display data in an interactive manner.

The module enables users to filter, sort, and drill down into specific subsets of the data, offering flexibility in analysis. By incorporating data visualizations such as bar charts, line graphs, pie charts, and scatter plots, users can quickly grasp key insights without having to interpret raw numbers.

The interactive nature of the dashboard allows for real-time updates as new data is collected, providing users with an up-to-date view of the metrics they care about.

This module is also highly customizable, allowing users to tailor the display to suit their needs, such as adjusting color schemes, data types, and layout. It supports integration with various data sources, such as databases or APIs, ensuring that it can work with data from multiple origins.


Fig- Interactive Dashboard & Visualization Module

**5. Containerization and Deployment Module**

The Containerization and Deployment Module ensures that the system's components can be packaged, deployed, and executed consistently across different environments, improving scalability, portability, and reliability. This module leverages containerization technologies, such as Docker, to

package the system's applications, dependencies, and libraries into isolated, self-contained units called containers. Each container encapsulates a specific function of the system, such as data collection, storage, or visualization, allowing the system to run efficiently regardless of the underlying hardware or operating system.

The module also manages the deployment of these containers to various environments, including local development, testing, staging, and production servers. By using container orchestration tools like Kubernetes, the module can automate the deployment process, scaling resources as necessary based on system demands.

Additionally, this module supports Continuous Integration and Continuous Deployment (CI/CD) pipelines, enabling automated testing, building, and deployment of the system's updates. It ensures that code changes are thoroughly tested before being deployed, reducing the risk of errors in production environments. The containerization approach also facilitates easy rollbacks in case of failures, as previous versions of the system can be quickly redeployed.

The flexibility provided by this module ensures that the system can be run on a variety of cloud platforms or on-premise servers, offering a wide range of deployment options. Overall, the Containerization and Deployment Module is crucial for the efficient, reliable, and scalable operation of the system across diverse environments.


Fig- Containerization & Deployment Module

**6. Cloud Hosting and Scalability Module**

The Cloud Hosting and Scalability Module is responsible for hosting the system on cloud platforms, providing flexibility, reliability, and the ability to scale resources based on demand. This module enables the system to take full advantage of cloud computing services, offering virtual machines, storage, networking, and other services that can be provisioned and managed dynamically. The module supports automatic scaling, allowing the system to handle varying loads by adjusting the number of resources allocated based on traffic or data processing requirements.

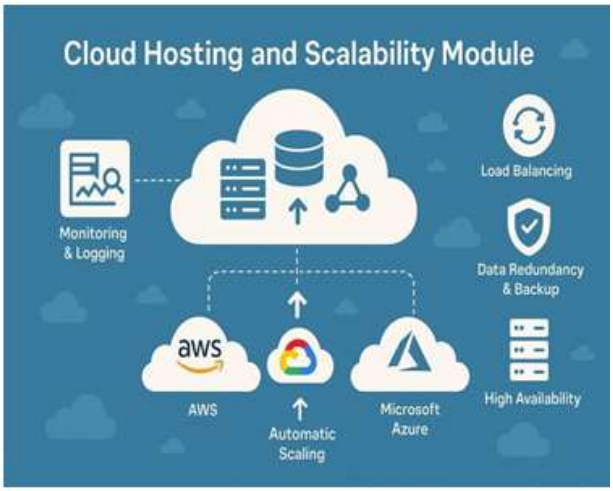


Fig- Cloud Hosting & Scalability Module

For instance, during periods of high demand, the system can scale out by deploying additional instances of services, and when demand decreases, it can scale down to reduce resource usage and cost. Cloud platforms like AWS, Google Cloud, or Microsoft Azure can be integrated, offering high availability, load balancing, and fault tolerance to ensure that the system remains operational even during hardware failures or outages. The module also provides features for data redundancy and backup, ensuring that critical information is replicated and stored securely across multiple regions. Additionally, this module can integrate with cloud-based monitoring and logging tools to track system performance and detect issues in real-time.

## IV. CONCLUSION

The integration of gamification in cybersecurity awareness training, as exemplified by the Zero-Day Game, represents a significant step forward in educating individuals about critical cyber threats and the best practices for mitigating them. Through interactive scenarios that simulate real-world challenges, players are encouraged to make informed decisions, which enhances their understanding of cybersecurity concepts in a practical, engaging way. The game's multi-level structure, featuring threats like weak passwords, phishing, cracked software, and social engineering, ensures a comprehensive exploration of cybersecurity issues, providing users with hands-on experience in a risk-free environment.

Ultimately, the Zero-Day Game demonstrates the potential of gamified learning to foster cybersecurity awareness. By making complex concepts accessible and engaging, it not only educates users but also empowers them to adopt better cybersecurity practices. As cyber threats continue to evolve, such innovative training solutions will be crucial in equipping individuals with the knowledge and skills necessary to protect themselves and their organizations from increasingly sophisticated attacks.

## REFERENCES

1. Mishra, P., et al. (2021). Automated YouTube Data Analysis Using API Services. Journal of Data Science and Analytics, 13(4), 256–264.
2. Singh, A., & Verma, R. (2022). Visualizing YouTube Trends using Python and Plotly. International Journal of Web Applications, 8(2), 101–110.
3. Kumar, S., et al. (2023). Streamlit-Powered Interactive Dashboards for Media Analytics. ACM SIGAPP Applied Computing Review, 23(1), 45–55.
4. Sharma, N., & Gupta, M. (2022). Containerizing Analytics Applications with Docker for Cloud Scalability. IEEE Cloud Computing, 9(3), 30–39.
5. Patel, R., et al. (2021). Cloud-Based Hosting for Real-Time Data Platforms. International Journal of Cloud Computing, 11(2), 97–108.
6. Rao, D., & Naik, K. (2020). Secure Firebase Authentication in Scalable Web Applications. Journal of Information Security and Applications, 53, 102521.
7. Jain, P., & Das, M. (2020). Authentication Frameworks in Cloud and Web Systems: A Firebase- Based Approach. Journal of Cybersecurity Technologies, 4(1), 33–44.
8. Ali, H., et al. (2023). API-Driven Real-Time Media Analytics with Embedded Cloud Dashboards. International Journal of Interactive Systems, 6(1), 59– 70.
9. Mehta, T., & Rao, A. (2024). Efficient Evaluation of Video Performance Metrics Using NoSQL Databases. Journal of Data Engineering and Visualization, 2(2), 112–121.
10. Bansal, A., & Iyer, V. (2019). Scalable Multimedia Analytics Pipelines Using Docker and Kubernetes. Journal of Multimedia Systems and Applications, 17(3), 88–98.