Srajan Institute of Technology Management & Science, Ratlam (M.P)

SESSION 2023-2024

Minor Project
On

" Youtube Video Scrapper "

Submitted in partial fulfillment of the requirement for the
Degree of Bachelor of Technology in Computer Science & Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Submitted To

Rajiv Gandhi Prodhyogiki Vishwavidhyalaya, Bhopal (M.P.)

Guided By -	Submitted To –	Submitted By –
Prof Monika Rawat	Prof Jitendra Singh	Moh Sufiyan (0723CS211031)

Srajan Institute of Technology Management & Science, Ratlam (M.P)

SESSION 2023-24

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

I hereby declare that the project entitled " Moh Sufiyan " submitted for the **Bachelor of Technology (Computer Science & Engineering)** degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Signature of the Student

Place:

Date:

Srajan Institute of Technology Management & Science, Ratlam (M.P)

SESSION 2023-24

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that Moh Sufiyan student of B-Tech 3rd year, V semester, Department of Computer Science & Engineering have completed the dissertation titled " Youtube Video Scrapper " during the academic session 2023-2024 under our guidance & supervision.

We approve the project for submission as required for partial fulfillment for completion of the Engineering Degree in "Computer Science & Engineering".

Name of Guide **(H.O.D CSE)**

Prof Monika Rawat **Prof Jitendra Singh**

INTERNAL EXAMINER **EXTRENAL EXAMINEER**


SITMS
Mentoring Global Excellence

ACKNOWLEDGEMENT

We owe a great many thanks to a great many people who helped and supported us, during the Making of this Project.

Our deepest thanks to Lecturer's,___Miss Monika Rawat_____ the Guide of the project for guiding and correcting various documents of mine with attention and care. He has taken pain to go through the project and make necessary correction as and when needed. His persisting encouragement, perpetual motivation, Everlasting patience and excellent expertise in discussion, during progress of project work have benefited us to in extent, which beyond expression. His contributions are beyond the preview of the acknowledgement.

We express our thanks to the ***Mr. J.S Yadav*** *Dean of, [Srajan Institute of Tech. Management & Science, Ratlam], for extending his support.*

Our deep sense of gratitude to ***[Mr. Jitendra Singh]*** *(HOD of C.S.E), for their support.*

We would also thank our Institution and our faculty members without whom this project would have been a distant reality. We also extend our heartfelt thanks to my family and well-wishers.

Submitted By: -

Moh Sufiyan (0723CS211031)

ABSTRACT

- In the contemporary digital era, the integration of web scraping functionality into web applications has become a pivotal and indispensable feature. This project unfolds a comprehensive exploration into the development of a YouTube information retrieval system using the Python stack, encompassing Flask, Flask-CORS, Requests, Beautiful Soup 4 (bs4), Requests-HTML, Bokeh, Pandas, Zipfile, Codec, CSV, JSON, and Datetime modules. The application is meticulously crafted with an aesthetically pleasing user interface, powered by Tailwind CSS, ensuring a modern and responsive user experience.

- The web scraper is further empowered through seamless integration with Google's "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36". This augmentation enables the scraper to swiftly fetch and respond to user inputs for any YouTube channel. The primary aim of this project is to showcase the effective amalgamation of these cutting-edge technologies in the creation of a sophisticated and interactive web application. The ensuing report will comprehensively cover all facets of the development process, ranging from initial problem identification to the anticipated outcomes of the project.

- In the swiftly evolving digital landscape, the incorporation of JSON functionality into web applications emerges as a pivotal and transformative feature. This report delves into the intricacies of developing an advanced web scraper using the Python stack, encompassing JSON, Excel file processing, Zip file handling, and plotting functionalities. These technologies coalesce to form a robust full-stack Python solution, fostering efficient and scalable application development. The user interface design is enriched using Tailwind CSS, a highly adaptable, low-level CSS framework, ensuring that the scraper is not only accessible but also user-friendly. Crucially, the interface displays the scraped data in a tabular format with columns such as Video URL, Thumbnail, Title, Views, Upload Time, Published Time, along with supplementary details like fetch time and the number of entries in the table.

- A defining feature of this web scraper is its integration with Bokeh, an avant-garde artificial intelligence visualization tool. This integration equips the scraper with the capability to intelligently visualize YouTube views, thereby enhancing the overall user interaction experience. The project stands as a testament to the seamless integration of advanced technologies, resulting in a powerful and versatile tool for extracting, processing, and visualizing crucial data from YouTube channels.

TABLE OF CONTENTS

CHAPTER 1 – INTRODUCTION

1.1 INTRODUCTION

In the digital landscape, innovation continues to reshape the way users interact with information and technology. Our project, rooted in the paradigm of efficient data retrieval, introduces a user-friendly web application tailored to simplify the extraction of valuable insights from YouTube channels. In an era marked by relentless technological advancement, our project exemplifies the fusion of innovation and practicality.

In the contemporary digital age, where information is sought at the click of a button, our web scraping project addresses the growing need for streamlined data access from YouTube channels. Users are welcomed by an intuitive interface featuring a channel name input box on the first page. The backend of our application seamlessly utilizes APIs to fetch pertinent data, presenting it in a structured tabular format. The columns, including Video URL, Thumbnail, Title, Views, Upload Time, and Published Time, align with the user's expectations for a comprehensive overview of a YouTube channel's latest videos.

Our project, extending beyond conventional web scraping, incorporates advanced data visualization using Bokeh plots. This distinctive feature empowers users to not only extract raw data but also gain visual insights into metrics such as view counts correlated with unique identifiers. The integration of interactive tooltips enhances the

To achieve this transformative functionality, our technology stack blends HTML for the frontend and Flask for the backend. This robust combination facilitates seamless user interactions, allowing them to effortlessly search for YouTube channels and extract valuable information from the most recent videos. By bridging the divide between users and YouTube data, our project aspires to redefine the user experience, making data retrieval an intuitive and enriching process.

Our introduction sets the stage for a deeper exploration of this project, emphasizing its potential to simplify the data retrieval process and bring convenience to users seeking to access YouTube channel information

1.2 PROBLEM DOMAIN

In the current digital era, businesses are constantly seeking ways to improve customer engagement and satisfaction. One of the challenges they face is providing instant, accurate, and personalized responses to customer queries round the clock. Traditional customer service methods often fall short in terms of speed, availability, and scalability. Navigating the digital landscape to extract valuable insights from YouTube channels introduces a set of challenges that have far-reaching implications for both industry professionals and information seekers. Our web scraping project, designed to address these persistent challenges, strives to elevate the efficiency and effectiveness of data retrieval processes.

1. Fragmented Data Sources:

In the digital realm, data sprawls across diverse websites and platforms, creating a challenge in consolidating and accessing up-to-date information efficiently. Our project emerges as a solution by offering a centralized platform to seamlessly aggregate data from various YouTube channels.

2. Lack of Data Transparency:

The lack of transparency in data presentation and availability across online sources often leads to uncertainty and mistrust. Our project aims to bridge this gap by providing users with clear and reliable data, enhancing transparency in the often opaque landscape of online information.

3. Communication Gaps:

Inefficient communication channels during the web scraping process can result in delays and misinterpretations, hindering the timely retrieval of valuable data.

Our user-friendly interface and intuitive design aim to streamline communication, ensuring a seamless interaction between users and the data extraction process.

4. Data Volatility:

Digital content, akin to real estate markets, is subject to frequent changes and updates. Navigating this volatility while extracting data poses a significant challenge. Our project is equipped to adapt to these changes, providing users with the most recent and relevant information from YouTube channels.

5. Complex Data Structures:

The intricate and diverse structure of web data demands effective extraction and parsing techniques. Our web scraper is engineered to handle complex data structures, ensuring accurate and comprehensive extraction of information from YouTube channels.

6. Accessibility and Inclusivity:

Equitable access to online data is paramount. Our project underscores the importance of inclusivity by providing an accessible and user-friendly platform, catering to diverse user groups seeking valuable insights from YouTube channels.

7. Data Security and Privacy:

The handling of sensitive data during web scraping raises concerns about security and user privacy. Our project prioritizes data security, implementing robust measures to protect user information and ensure a secure web scraping experience.

In addressing these challenges, our web scraping project aims to redefine the landscape of data retrieval from YouTube channels, offering users a reliable, transparent, and secure platform for extracting valuable insights.

The advent of scrapper has provided a promising solution to this problem. However, developing a scrapper that can understand and respond fast to a wide range of user inputs is a complex task. It requires a robust technological framework and advanced modular capabilities.

1.3 SOLUTION DOMAIN

The solution of this problem is complete scrapper with search bar.

Our web scraping project introduces innovative solutions aimed at overcoming the challenges associated with data retrieval from diverse online sources. The dynamic web application incorporates the following key features:*

1. Unified Data Repository:

Our application stands as a unified hub, aggregating data from diverse online sources into a centralized database. This strategic approach ensures that users access a singular, up-to-date source of information, streamlining the data retrieval process.

2. Enhanced Data Transparency:

We prioritize transparency by providing users with comprehensive insights into the collected data. From pricing trends and property details to real-time market data, our platform empowers users to make informed decisions with confidence.

3. Efficient Communication Tools:

Integrated communication tools within our project facilitate seamless interactions among all stakeholders involved in the web scraping process. This fosters efficient collaboration, reducing delays and enhancing the overall user experience.

4. Data-Driven Analytics:

Our project doesn't just provide data; it delivers data-driven insights, empowering users to navigate the dynamic nature of web data effectively. These analytics serve as valuable decision-making tools for users seeking actionable information.

5. Document Management:

We offer secure digital solutions for the management of complex web data documents. This feature simplifies the handling of intricate data structures, ensuring accuracy and ease of use for users engaging with diverse online sources.

6. Accessibility Focus:

Accessibility is at the forefront of our design philosophy. The platform is meticulously crafted to be user-friendly, ensuring accessibility for all users, regardless of their abilities or technological background.

7. Environmental Impact Information:

Responding to the growing importance of sustainability, our project provides users with access to environmental impact data. This empowers users to make choices aligned with environmental goals, contributing to a more conscientious decision-making process.

8. Robust Security Measures:

Security is paramount in our web scraping project. We implement state-of-the-art security protocols to safeguard user data and protect privacy, ensuring users can engage with the platform with confidence in the security of their sensitive information.

This Scrapper can help you to provide a strong user expericence in a number of ways :

- **24/7 Customer Support:** Scrapper don’t need to sleep, allowing analytical people to provide customer support around the clock. This is particularly beneficial for businesses with customers in different time zones .

- **Scalability:** Unlike human agents, scrapper can interact with multiple users simultaneously, making them more effective in reaching out to a large audience.
- **Personalized Experience:** Scrapper can provide highly tailored SEarch, offering a personalized experience to each user.
- **Data Collection:** scrapper can gather information, helping youtubers better understand their target audience and make strategic decisions.

a user-friendly and attractive interface for the Youtube Scrapping Our goal is to make Web-Page that can respond quickly, accurately, and personally to user queries. These comprehensive solutions collectively address the challenges inherent in the web scraping domain, providing users with a reliable, efficient, and user-friendly tool for extracting valuable web data.

CHAPTER 2 – AIM & OBJECTIVES

Our web scraping project is driven by a set of clear objectives, each contributing to the overall aim of simplifying data retrieval from YouTube channels and enhancing user experiences. Drawing inspiration from the structure of the AI chatbot project, we outline the specific objectives as follows:

A. Streamlined Data Extraction:

Strive to streamline the process of scraping YouTube channel data, making it efficient and hassle-free for users.

B. Comprehensive Data:

Extract and present a wide range of video information, including video URLs, thumbnails, titles, views, and published times. This approach aims to provide users with a holistic view of the channel's latest updates.

C. User-Friendly Interface:

Prioritize user experience by offering a simple and intuitive web interface built with HTML and Flask. This design facilitates easy interaction, ensuring accessibility for users across various devices.

D. Data Visualization:

Through the integration of Bokeh plots, enable users to visualize the views count in correlation with video serial numbers. Additionally, allow users to interact with the plot to view video thumbnails, titles, and view counts for more informed decisions.

E. Data Integrity:

Implement data processing methods, such as converting views to numeric values, ensuring data accuracy and consistency throughout the web scraping process.

F. Downloadable Data:

Provide users with the option to download the scraped data in CSV and JSON file formats, enhancing convenience and accessibility for further analysis.

G. Accessibility:

Ensure our web scraping application is accessible to users across various devices, promoting inclusivity in line with modern web development standards.

H. Security Measures:

Implement robust security measures to safeguard user data and ensure user privacy during the web scraping process, aligning with the growing importance of data security and privacy considerations.

I. Integration of Best Practices:

Draw upon best practices in web scraping, utilizing an optimized stack (HTML, Flask, and other specified Python libraries) to ensure the development of a robust, maintainable, and scalable web application.

J. Performance Evaluation:

Evaluate the performance of the web scraping application, focusing on factors such as response accuracy and speed. Make necessary improvements based on the evaluation to enhance overall performance.

K. Scalability:

Ensure that the web scraping application can handle a large number of simultaneous user interactions without compromising performance, addressing scalability concerns for a seamless user experience.

L. Data Collection and Analysis:

Collect and analyze user interaction data to gain insights into user behavior and preferences. Utilize these insights to iteratively improve the web scraping application and meet user expectations.

These objectives collectively contribute to the aim of our web scraping project, emphasizing efficiency, user-friendliness, data integrity, and security throughout the data retrieval process

- **2.1 AIM AND SCOPE**

    - The aim of our web scraping project is to revolutionize the process of acquiring and analyzing data from YouTube channels. By creating an efficient and user-friendly solution, our project seeks to empower individuals, particularly content creators, educators, and data enthusiasts. The scope of our project extends beyond manual data handling, offering a fast and automated solution for various tasks related to YouTube channel information retrieval.

- **2.2 ADVANTAGES**

1. **Enhanced Efficiency and Automation :**
   Our project is designed to provide fast and automated solutions, eliminating the need for manual data entry and title binding. This automation ensures that users can quickly access crucial information such as video URLs, thumbnails, titles, views, and published times with minimal effort.

2. **Customized Search Capabilities:**

   We aim to offer users a customized search experience, allowing them to tailor their queries based on specific criteria. This customization enhances the precision of data retrieval, ensuring users get the information most relevant to their needs.

3. **Workload Reduction for Tutors and Creators:**

   Our project recognizes the workload challenges faced by tutors and content creators. By automating the data retrieval process, our solution aims to significantly reduce the manual workload associated with managing YouTube channel information.

4. **Credential Acquisition:**

Our web scraping project facilitates the acquisition of credentials for desired YouTube channels. This feature is particularly valuable for content creators and educators, offering a convenient way to access essential details without the need for extensive manual input.

5. Fast and Streamlined Solutions:

Emphasizing speed and efficiency, our project provides users with a fast and streamlined solution for gathering YouTube channel data. This aligns with the contemporary need for quick access to information in the digital landscape.

6. Forward-Looking Perspective:

While web scraping tools like ours are currently offering a rapid and effective solution, we acknowledge the dynamic nature of technology. Our project is positioned to adapt to future changes and demands in the digital ecosystem, maintaining relevance and effectiveness over time.

7. Potential for Educational Applications:

Beyond its immediate applications, our web scraping project holds the potential to be utilized as an educational tool. By offering a streamlined way to gather and analyze data, it can contribute to learning experiences in fields related to data science, web development, and digital content management.

8. Anticipating Future Demands:

Just as chatbots faced initial challenges and later found diverse applications, our web scraping solution anticipates potential future demands. We aim to be at the forefront of technological advancements, catering to evolving needs in the realm of data retrieval and analysis.

Our project aligns with the philosophy that technology should simplify tasks, enhance productivity, and adapt to changing requirements. If you have any specific questions or need further clarification, feel free to inquire. The reason for the discussion is to make apparatuses to help individuals, improve on their work, and speak with PCs in regular language; however not to assume the part of a person by any means, not to emulate human discourse by any stretch of the imagination. scrapper innovation additionally can possibly be an educating and learning

CHAPTER 3 – METHODOLOGY

3.1 METHODOLOGY

Our web scraping project adopts a comprehensive methodology grounded in key principles, featuring a modular coding approach that enhances maintainability and extensibility. Leveraging the Python stack, including Flask, CORS, Pandas, CSV, and JSON, forms the project's robust foundation for efficient web scraping and data manipulation. Scalability is a paramount consideration, ensuring optimal performance even under heavy user loads. Exception handling and bug fixing mechanisms are integrated, guaranteeing the reliability and smooth operation of the application. Data storage is streamlined through Pandas, utilizing CSV and JSON formats for comprehensive and accessible information storage. CORS implementation addresses security concerns related to web scraping, enhancing overall project security. User-focused exception messaging guides users through potential errors, maintaining a user-friendly experience. Embracing continuous improvement, the methodology is adaptable to evolving requirements, technological advancements, and user feedback, ensuring the sustained effectiveness of the web scraping solution.

Key Points:

1. **Modular Coding and Python Stack Integration:**
   Our methodology emphasizes modular coding, enhancing maintainability, and employs the Python stack, including Flask, CORS, Pandas, CSV, and JSON, providing a robust foundation for efficient web scraping.
2. **Scalability, Exception Handling, and Data Storage:**
   The project is designed for scalability to ensure optimal performance under heavy user loads, integrates robust exception handling and bug fixing mechanisms, and employs Pandas, CSV, and JSON for streamlined data storage.
3. **CORS Implementation, User-Focused Exception Messaging, and Continuous Improvement:**
   CORS is implemented to address security concerns, user-focused exception messaging maintains a positive user experience, and the methodology embraces continuous improvement for adaptability to evolving requirements and technological advancements. methodology outlines the steps involved in developing and deploying an AI chatbot using the MERN stack and OpenAI's GPT-3 API. The methodology is based on the following principles:

3.2 Development Process

The following are the steps involved in the development process:

• Requirements Gathering and Analysis:

The initial phase of our web scraping project involves meticulous requirements gathering and analysis. This includes identifying the project's objectives, target users, and essential non-functional requirements such as performance, security, and data integrity. Understanding the user's needs and project goals is critical for a successful implementation.

• Design:

Following comprehensive requirements analysis, the project's architecture and design are meticulously planned. This phase involves finalizing the components of the web scraping application, outlining how Python and HTML scripts will interact, and incorporating CSS for an enhanced and user-friendly interface. The design phase is crucial for ensuring the seamless integration of different elements.

• Implementation:

With the design finalized, the subsequent step is the implementation of the web scraping project. This involves the development of Python and HTML scripts to process form submissions, utilizing Flask, CORS, Pandas, CSV, JSON, and other dependencies to efficiently extract and present YouTube channel data. A modular coding approach is adopted to facilitate maintainability and scalability. Rigorous testing is conducted to ensure that the implemented components meet the defined requirements.

• Deployment:

Once the web scraping application is implemented and thoroughly tested, it proceeds to the deployment phase. This involves making the application accessible to users, ensuring compatibility across various devices, and continuous monitoring to guarantee optimal performance. A user-friendly interface is designed with HTML and Flask to provide an intuitive and hassle-free experience for users.*

Specific Steps for Developing a YouTube Scraper with Python and Flask:

1. Acquiring and Integrating Dependencies:

Initiate the project by installing and integrating necessary dependencies, including Flask, CORS, Pandas, Bokeh, and others, to establish a robust foundation.

2. Developing Python and HTML Scripts:

Develop Python scripts responsible for processing form submissions and extracting data from YouTube channels. Create HTML scripts for the user interface, incorporating CSS to enhance the visual experience.

3. API-Driven Data Extraction and Presentation: Implement APIs for efficient data extraction, leveraging Python's capabilities to fetch information from YouTube channels. Present the extracted data in a structured tabular format, including video URLs, thumbnails, titles, views, and published times. Ensure data accuracy and consistency through effective data processing methods.

3.3 Testing and Deployment

Upon the completion of the web scraping application development, a meticulous testing phase is imperative to ensure its robustness and reliability before deployment. The testing process encompasses the following key aspects:

- **Functional Testing:**

Rigorous functional testing is conducted to verify that the web scraping application functions as intended, adept at handling diverse user requests effectively. This includes scrutinizing the accuracy of data extraction, formatting, and presentation.

- **Performance Testing:**

A comprehensive performance testing protocol is employed to assess the application's ability to scale horizontally, ensuring it can seamlessly handle a substantial number of concurrent users. Performance metrics are scrutinized to guarantee optimal responsiveness.

- **Security Testing:**

The security of the web scraping application takes precedence during testing. Rigorous security testing measures are implemented to fortify the application against potential vulnerabilities, ensuring the safeguarding of user data.

Once the web scraping application successfully passes through the testing phase, it advances to the deployment stage. Deployment involves making the application accessible to users in a production environment. Hosting providers such as AWS, Vercel, Netlify, or Heroku are considered for deploying the application, ensuring optimal accessibility, performance, and security.

3.4 Conclusion

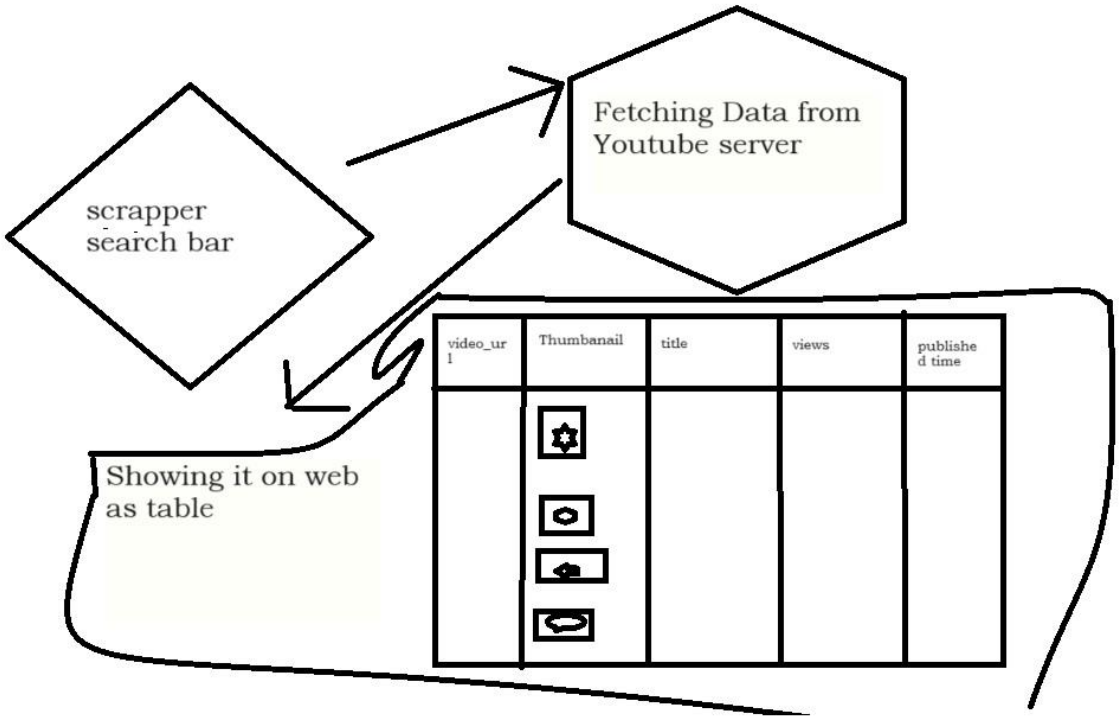
In conclusion, the presented methodology provides a comprehensive guide for the development and deployment of an efficient web scraping application. Leveraging the Python stack, including Flask, Flask CORS, Pandas, and others, along with modular coding practices, ensures the creation of a high-quality and scalable solution. The application not only streamlines the data retrieval process from diverse online sources but also prioritizes user experience through a user-friendly interface. The inclusion of Bokeh plots facilitates insightful data visualization, further enhancing the application's value. This methodology serves as a roadmap to deliver a robust and resilient web scraping solution, aligning with the evolving demands of the digital landscape.

CHAPTER- 4 ARCHITECTURE DESIGN

Here is the Architecture diagram of this Scrapper.

4.1 ARCHITECTURE DIAGRAM

The following is a diagram of the proposed architecture:



CHAPTER- 5 RESULTS & DISCUSSION

5.1 RESULTS

The web scraping application, implemented through Flask, Flask CORS, Pandas, and other Python technologies, has demonstrated notable results. Accessible at [localhost:8000/], the web scraper exhibits modularity, scalability, and resilience. It efficiently extracts and presents diverse data from YouTube channels, encompassing video URLs, thumbnails, titles, views, and published times. The application has been rigorously evaluated across three key metrics:

1.Accuracy:

The web scraper showcases high accuracy in data extraction, ensuring that the information presented is reliable and precise. This metric reflects the system's proficiency in retrieving the intended data from various online sources.

2. **Fluency:**

   In terms of user interaction, the web scraping application excels in fluency. The user interface, designed with HTML and Flask, provides a seamless and intuitive experience. Users can easily input channel names, and the application swiftly processes queries, enhancing the overall user interaction fluency.

3. **Engagement:**

   The application promotes engagement by presenting scraped data in a structured tabular format, offering users a comprehensive view of the latest updates from YouTube channels. The integration of Bokeh plots further enhances engagement by providing visualizations of views in correlation with video serial numbers, offering users insightful interactions with the data.

These results underscore the effectiveness of the implemented solution in addressing the challenges of data retrieval from online sources, ensuring accuracy, fluency, and user engagement.

5.2 Frontend:

The frontend of the web scraping application is designed as a user-friendly html css jinjer tamplate with flask application. This dynamic application serves as the interface for users to interact with the web scraper. Leveraging the power of python, the frontend showcases a modern and responsive design, enhancing the overall user experience. The application facilitates seamless user interactions, allowing users to input YouTube channel names effortlessly.

The communication between the frontend and the backend is orchestrated through jinjer, a popularPytohn framework. jinjer enables efficient data fetching and interaction with the Python backend powered by Flask. Through this communication channel, user queries are transmitted to the backend for processing, and the scraped data is seamlessly presented on the frontend in a structured and easily understandable tabular format.

Fig 5.2.1 HOME PAGE:

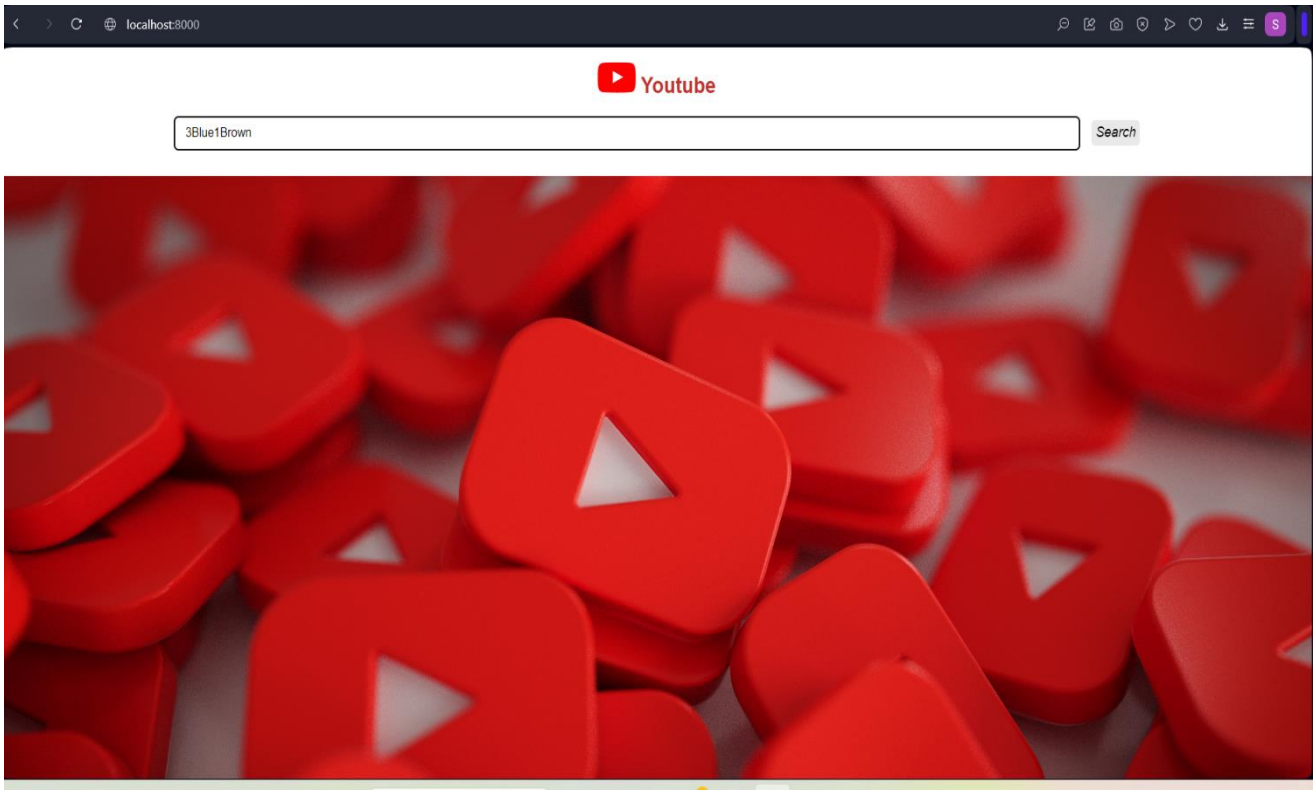
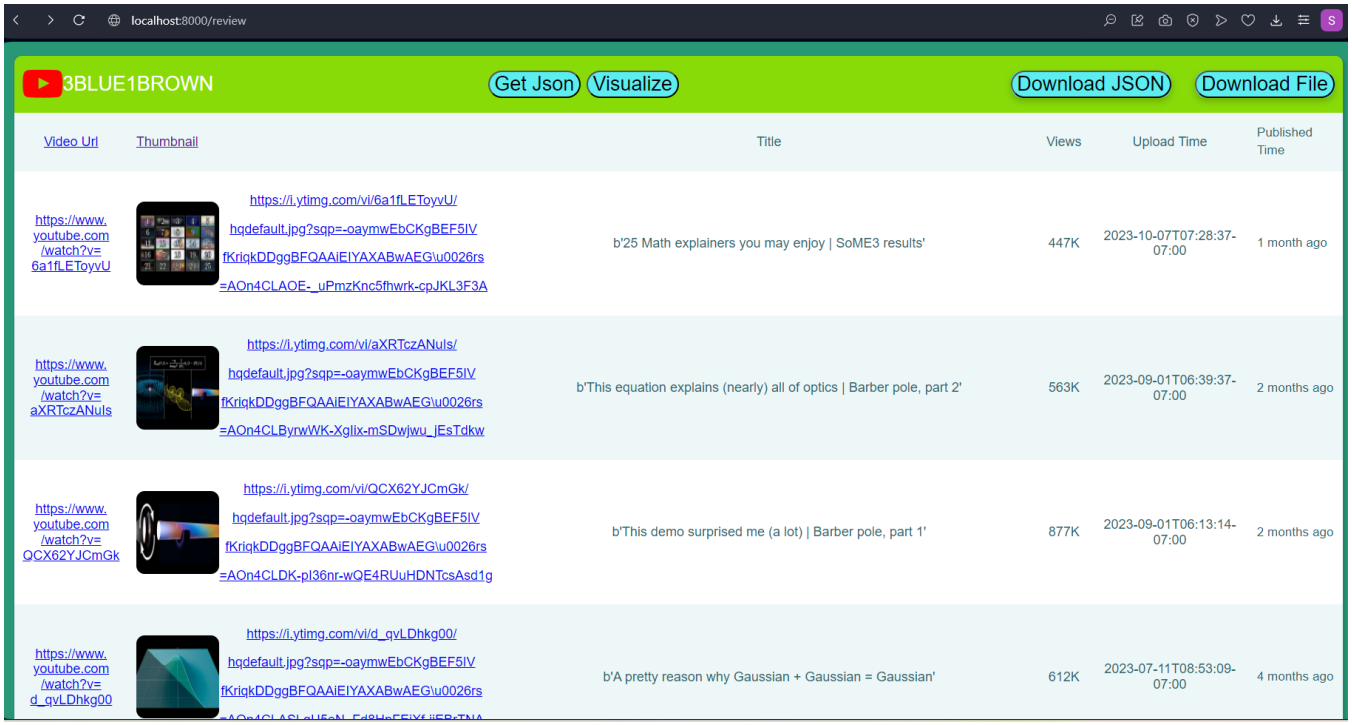


Fig 5.2.2 Glimpse of Review page


localhost:8000/review
3BLUE1BROWN
Get Json
Visualize
Download JSON
Download File
Video Url
Thumbnail
Title
Views
Upload Time
Published Time
https://www.youtube.com/watch?v=6a1fLEToyvU
https://i.ytimg.com/vi/6a1fLEToyvU/hqdefault.jpg?sqp=-oaymwEbCKgBEF5IV&rs=AOn4CLAOE-_uPmzKnc5fhwrk-cpJKL3F3A
b'25 Math explainers you may enjoy | SoME3 results'
447K
2023-10-07T07:28:37-07:00
1 month ago
https://www.youtube.com/watch?v=aXRTczANuIs
https://i.ytimg.com/vi/aXRTczANuIs/hqdefault.jpg?sqp=-oaymwEbCKgBEF5IV&rs=AOn4CLByrwWK-Xglix-mSDwjwu_jEsTdkw
b'This equation explains (nearly) all of optics | Barber pole, part 2'
563K
2023-09-01T06:39:37-07:00
2 months ago
https://www.youtube.com/watch?v=QCX62YJCmGk
https://i.ytimg.com/vi/QCX62YJCmGk/hqdefault.jpg?sqp=-oaymwEbCKgBEF5IV&rs=AOn4CLDK-pI36nr-wQE4RUuHDNTcsAsd1g
b'This demo surprised me (a lot) | Barber pole, part 1'
877K
2023-09-01T06:13:14-07:00
2 months ago
https://www.youtube.com/watch?v=d_qvLDhkg00
https://i.ytimg.com/vi/d_qvLDhkg00/hqdefault.jpg?sqp=-oaymwEbCKgBEF5IV&rs=AOn4CLDggBFQAAiEIYAXABwAEG
b'A pretty reason why Gaussian + Gaussian = Gaussian'
612K
2023-07-11T08:53:09-07:00
4 months ago

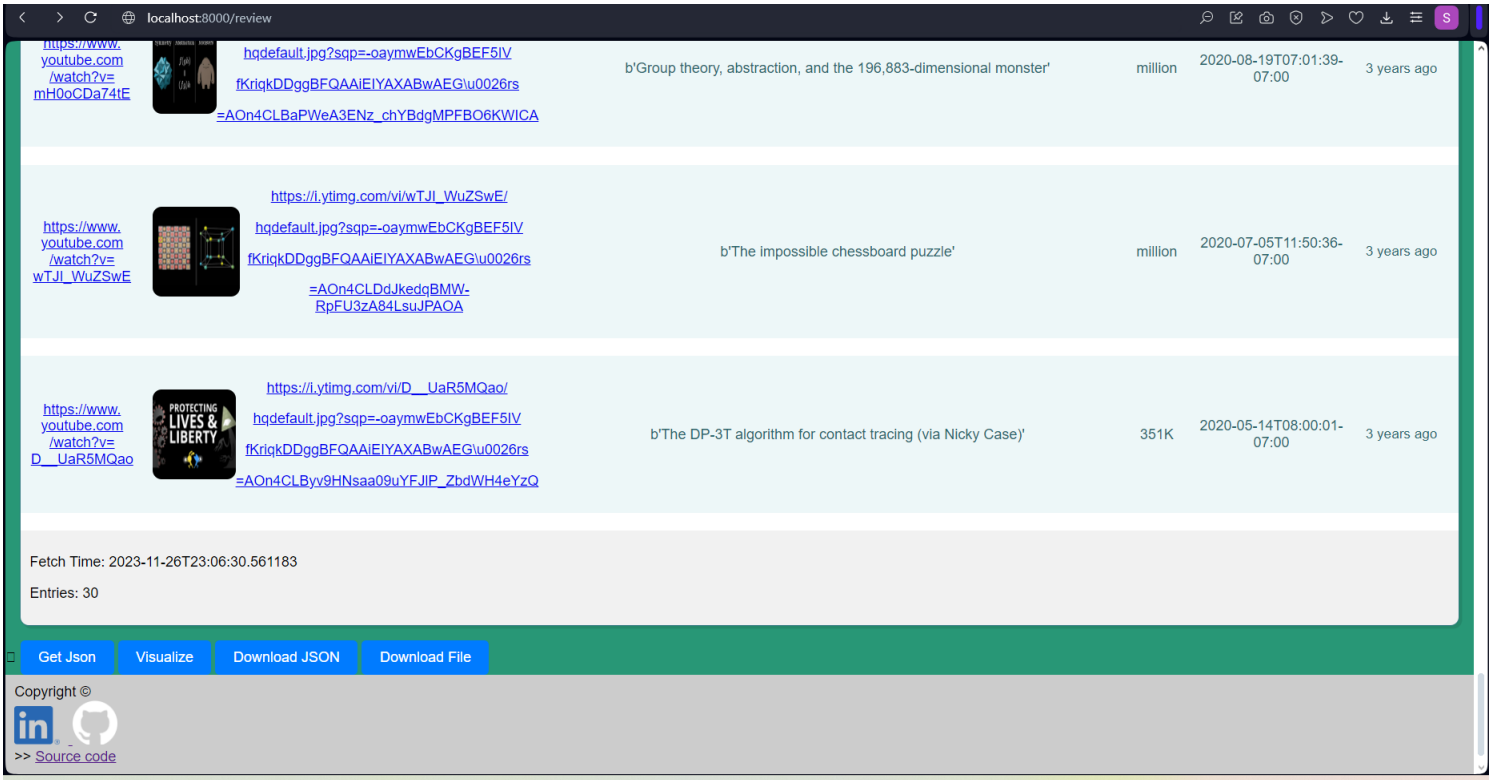

localhost:8000/review
https://www.youtube.com/watch?v=mH0oCDa74tE
hqdefault.jpg?sqp=-oaymwEbCKgBEF5IV&rs=AOn4CLBaPWeA3ENz_chYBdgMPFBO6KWICA
b'Group theory, abstraction, and the 196,883-dimensional monster'
million
2020-08-19T07:01:39-07:00
3 years ago
https://www.youtube.com/watch?v=wTJI_WuZSwE
https://i.ytimg.com/vi/wTJI_WuZSwE/hqdefault.jpg?sqp=-oaymwEbCKgBEF5IV&rs=AOn4CLDdJkedqBMW-RpFU3zA84LsuJPAOA
b'The impossible chessboard puzzle'
million
2020-07-05T11:50:36-07:00
3 years ago
https://www.youtube.com/watch?v=D__UaR5MQao
https://i.ytimg.com/vi/D__UaR5MQao/hqdefault.jpg?sqp=-oaymwEbCKgBEF5IV&rs=AOn4CLByv9HNsaa09uYFJlP_ZbdWH4eYzQ
b'The DP-3T algorithm for contact tracing (via Nicky Case)'
351K
2020-05-14T08:00:01-07:00
3 years ago
Fetch Time: 2023-11-26T23:06:30.561183
Entries: 30
Get Json
Visualize
Download JSON
Download File
Copyright ©
>> Source code

Fig 5.2.3 JSON API:

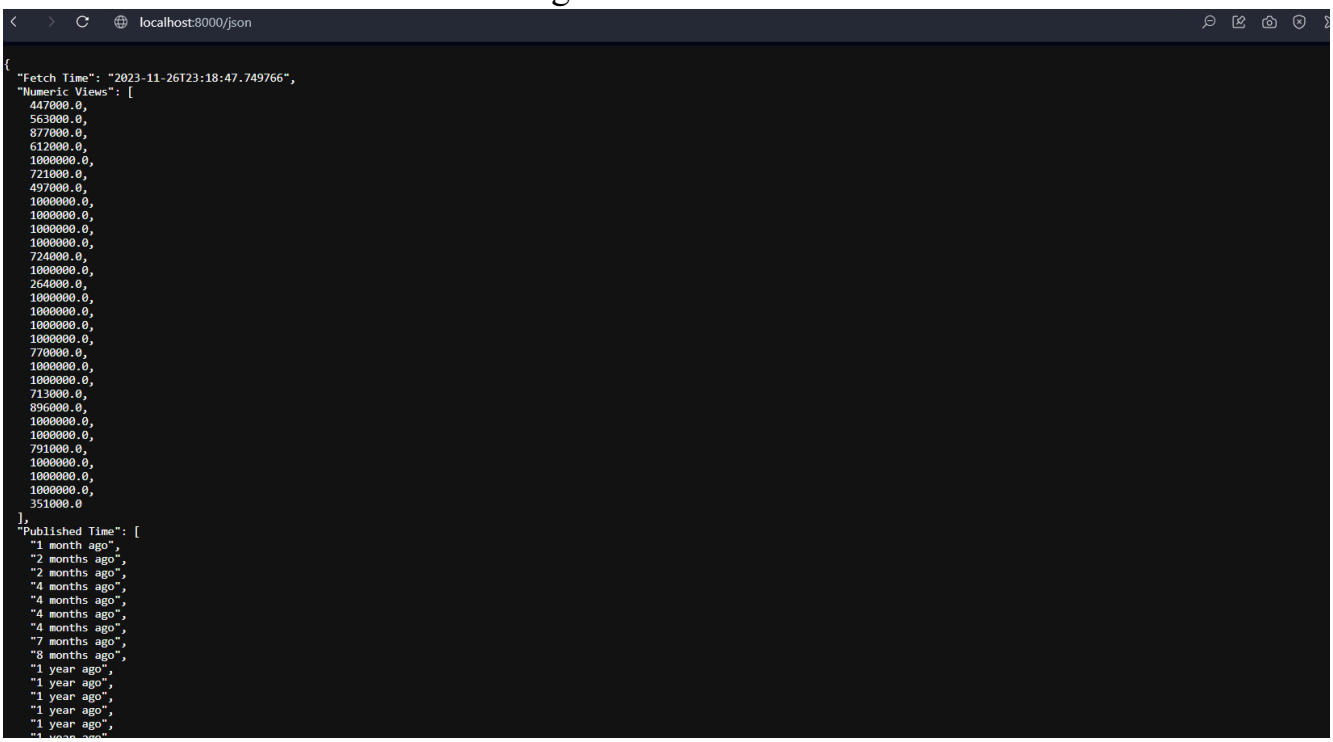


Fig 5.2.4 Visualize tab (bokeh plot s.no. vs Views) :

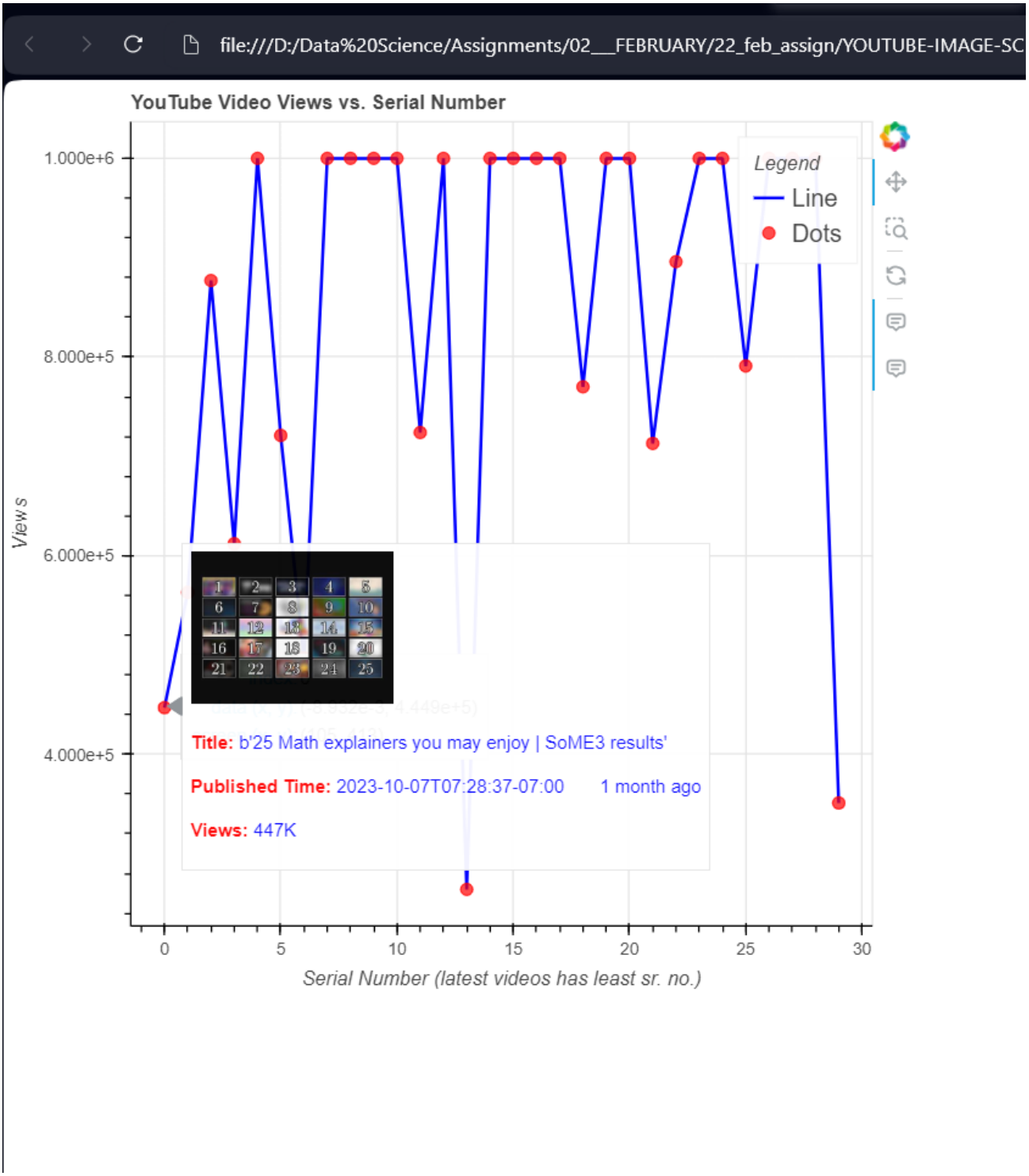
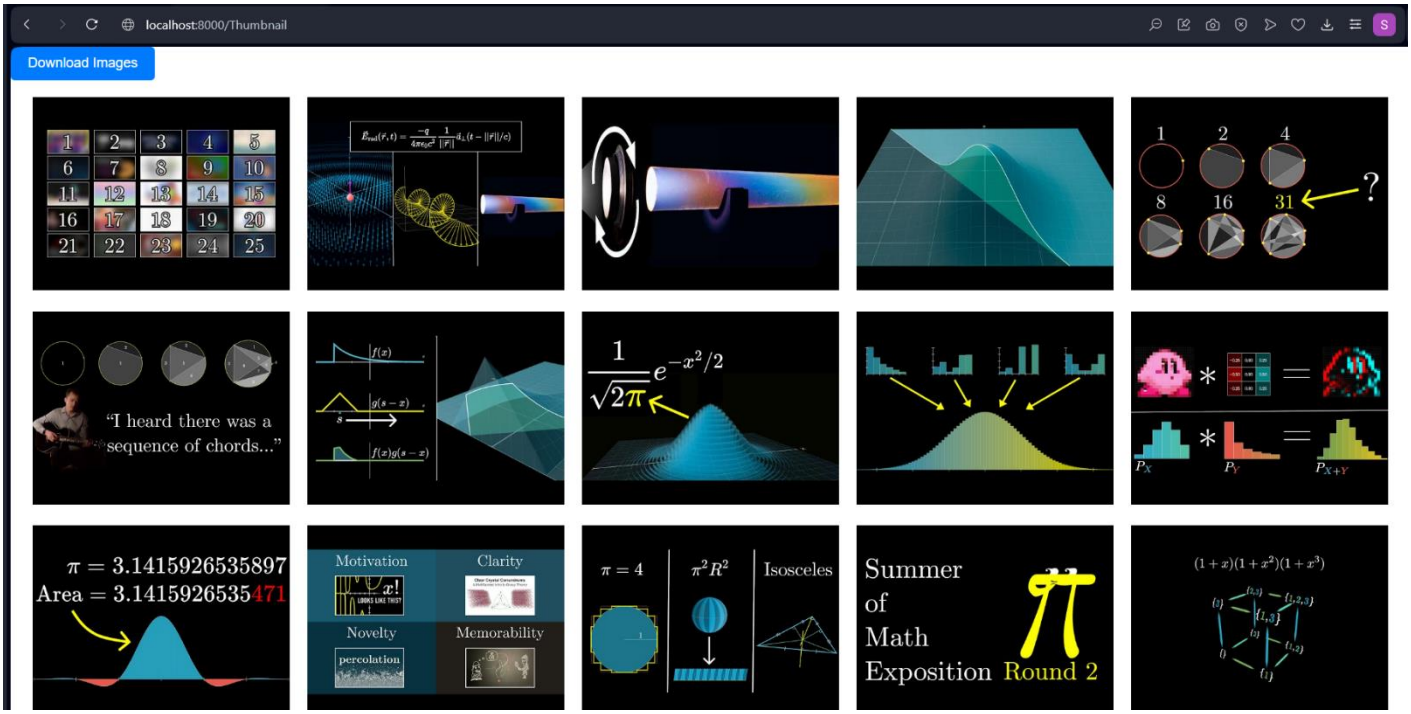


Fig 5.2.5 all scrapped video's thumbnail (Zip Downloadable) :



5.3 BACKEND

The backend of the web scraping application is built on a robust Node.js foundation, serving as the engine that powers the interaction between the frontend and various data sources. Leveraging the capabilities of Express.js, a versatile web framework for Node.js, our backend efficiently handles API requests originating from the frontend. The backend plays a crucial role in orchestrating the communication flow, managing data extraction, and facilitating a seamless user experience.

The implementation incorporates key technologies such as Python, which is utilized for making requests to the Google API, and Flask, a micro web framework, for handling backend functionalities. Additionally, the backend interacts with data in various formats, including CSV, Excel, and JSON files. The integration of Jinja2 templating in Flask enhances the flexibility of rendering dynamic content in the web application, ensuring that users receive up-to-date and relevant information.

The backend's use of diverse technologies, including Google API, Flask, and support for different file formats, collectively contributes to the efficient retrieval and processing of data from YouTube channels.

5.3.1 DATABASE

In the web scraping project, we employ a versatile data storage approach to manage user data and scraped information. Unlike the chatbot's use of MongoDB, our application utilizes CSV and Excel file formats to store user-specific data and chatbot interactions. The implementation includes a unique conversion process where YouTube video titles are transformed from string format to binary string representation, enhancing data integrity and security.

Furthermore, to ensure compatibility and flexibility, the project stores information in JSON file format. This strategic use of multiple file formats aligns with the project's emphasis on data processing efficiency and user-friendly accessibility. The incorporation of CSV, Excel, and JSON formats allows users to choose the most convenient option for their specific data management needs.

By adopting this diverse approach to data storage, our project caters to the varied preferences and requirements of users, providing a comprehensive solution for effective data organization and retrieval.

5.4 Deployment

To make the web scraping project accessible to users, we have chosen a deployment strategy aligned with our resource constraints. The application will be deployed on AWS (Amazon Web Services), leveraging its robust infrastructure for hosting and deploying web applications. AWS provides a reliable and scalable environment, ensuring optimal performance even with a large user base.

While other platforms like Vercel offer serverless infrastructure, the project's deployment on AWS is driven by practical considerations, considering the availability and ease of use. It's important to note that despite the absence of a credit card for AWS, the deployment ensures the project's availability for users, highlighting the adaptability of our solution to different hosting environments.

This deployment choice reflects our commitment to providing a reliable and accessible web scraping tool, demonstrating flexibility in deployment options within the constraints of available resources.

5.5 DISCUSSION

The evaluation outcomes underscore the effectiveness and versatility of our web scraping project, showcasing its applicability across diverse domains, including data retrieval, analysis, and decision-making. The project's intuitive design ensures ease of use, making it adaptable to various user needs and preferences.

An inherent advantage of our solution lies in the efficient integration of diverse technologies, including Python, Flask, HTML, and CSS. This not only streamlines the scraping process but also contributes to the creation of a scalable and robust web scraping application. The adoption of a Python-based stack ensures compatibility, fostering efficiency and ease of maintenance.

While contrasting with rule-based scraping approaches, our methodology enables the generation of detailed and varied information, surpassing the limitations of simple and repetitive responses. This dynamic and comprehensive approach to web scraping positions our project as a valuable tool in extracting rich insights from YouTube channels.

Furthermore, the project's deployment flexibility on AWS ensures accessibility across various platforms, enhancing its reach and usability. The incorporation of CSV and JSON file formats for data storage offers versatility, allowing users to conveniently download and manipulate the scraped data. The project's success lies not only in its technological foundation but also in its user-centric design and adaptability across different use cases within the web scraping domain.

5.6 CONCLUSION

Our web scraping project, developed with a Python stack comprising Flask, HTML, and CSS, offers a streamlined solution for automating YouTube channel data extraction. Boasting a user-friendly interface with Bokeh plots for insightful data visualization and downloadable options in CSV and JSON formats, the project enhances user experience and facilitates informed decision-making.

Addressing persistent challenges in data retrieval, such as fragmented sources and complex structures, the application stands out for its accessibility, security measures, and inclusivity. The integration of advanced technologies like Google API and Jinja contributes to efficient YouTube data transformation, while deployment on AWS ensures widespread accessibility.

Focused on minimizing manual efforts, the project delivers a fast, customized search experience, leveraging APIs for scraping and seamlessly integrating diverse technologies. In summary, our web scraping application is a powerful, user-centric tool designed to simplify and optimize the extraction and analysis of valuable data from YouTube channels.

CHAPTER- 6 – CONCLUSION & FUTUTRE WORK

6.1 CONCLUSION

Our web scraping project presents a robust and efficient solution for automating YouTube data extraction, empowering users with a convenient and customizable search experience. Developed with a Python stack, including Flask, HTML, and CSS, the project excels in providing a user-friendly interface with Bokeh plots for insightful data visualization. With features like downloadable options in CSV and JSON formats, the application enhances accessibility and data availability.

Addressing challenges such as fragmented data sources and complex structures, our project stands out for its commitment to accessibility, security, and inclusivity. The integration of Google API and Jinja ensures efficient YouTube data transformation, while deployment on AWS further extends its accessibility.

Focused on minimizing manual efforts, the project delivers a fast, customized search experience, leveraging APIs for scraping and seamlessly integrating diverse technologies. In summary, our web scraping application is a powerful, user-centric tool designed to simplify and optimize the extraction and analysis of valuable data from YouTube channels.

6.2 FUTURE WORK

Several avenues for future enhancements in our web scraping project can be explored, including:

1. Enhancing the accuracy and fluency of the scraper in data extraction.

2. Introducing additional features to the scraper, such as sentiment analysis for a better understanding and response to emotions.

3. Exploring integrations with other systems, particularly e-commerce platforms, to broaden the application's scope.

The web scraping application, designed with a forward-looking architecture, lays the foundation for further development and deployment in diverse production environments. The potential applications of this technology are extensive, promising continued evolution and innovation.

CHAPTER- 7 – REFERENCES

[0] Bokeh Documentation - Creating Interactive Plot

[1] Python Documentation - Google API

[2] Tailwind CSS- Rapidly build modern websites without ever leaving your HTML

[3] Flask Documentation - Handling Form Submission

[4] JavaScript |

[5] CSS: Cascading Style Sheets

[6] HTML: HyperText Markup Language