iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** III **Month of publication:** March 2024

DOI: https://doi.org/10.22214/ijraset.2024.59536

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

Enhancing Collaboration and Efficiency through Automated Documentation of GCP Stored Procedures with genAI

Srilakshmi S
Data Engineer, Comcast – Sky, Chennai, India

Abstract: ***In the realm of cloud-based data management on the Google Cloud Platform (GCP), effective documentation of work processes is essential for streamlined operations. However, existing practices often rely on manual efforts, leading to incomplete and time-consuming documentation. To overcome this challenge, we propose a Lab Week project leveraging genAI's natural language processing capabilities. Our objective is to develop a Python script that automates the generation of comprehensive documentation for GCP Stored Procedures. This innovative solution aims to extract crucial details such as Common Table Expressions (CTEs), temporary tables, end tables, and associated columns, resulting in a structured Word document. By harnessing genAI's power, our automated documentation system not only saves time but also ensures consistency and depth in GCP Design Details. This initiative fosters collaboration and knowledge transfer among team members, with the generated Word documents seamlessly integrating into Confluence pages as valuable resources for our development teams.***
Keywords: Machine Learning, Generative AI, Vertex AI, Artificial Intelligence, Google Cloud Platform, Common Table Expressions, Graphical User Interface (GUI).

I. INTRODUCTION

In the fast-paced realm of cloud-based data management on the Google Cloud Platform (GCP), the documentation of work processes stands as a cornerstone for organizational success. However, our current practices in this domain fall short, relying heavily on manual efforts that often result in incomplete and time-consuming documentation. Recognizing the critical need for improvement, we propose a transformative Lab Week project powered by genAI's cutting-edge natural language processing capabilities. Our objective is clear: to revolutionize the documentation process for GCP Stored Procedures through automation. Stored Procedures serve as the backbone of our data operations on GCP, encapsulating intricate logic and data transformations. Yet, extracting meaningful insights from these procedures has traditionally been a labor-intensive task. With genAI, we aim to change that.

By developing a Python script that seamlessly integrates genAI's capabilities, we envision a paradigm shift in our documentation practices. This script will intelligently parse through Stored Procedures, extracting vital details such as Common Table Expressions (CTEs), temporary tables, end tables, and associated columns. The result? A meticulously structured Word document that encapsulates the essence of our GCP Design Our automated documentation system promises to foster collaboration and knowledge transfer among team members. With consistently formatted and comprehensive documentation at our fingertips, we empower developers, analysts, and stakeholders to make informed decisions and drive innovation with confidence.

Furthermore, by seamlessly integrating the generated Word documents into our Confluence pages, we establish a central repository of knowledge that serves as a beacon of clarity and efficiency for our development teams. This initiative represents more than just a technological advancement; it embodies our commitment to excellence, innovation, and continuous improvement in the realm of cloud-based data management. Through automation, we strive to unlock new levels of productivity, collaboration, and insight, positioning our organization at the forefront of GCP innovation such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

II. ARCHITECTURE

A. Google Cloud Platform Authentication

1) The user obtains a service account key file (key.json) from the GCP Console, which contains the necessary credentials for accessing GCP APIs programmatically.

2) The key.json file is securely stored and used for authentication during the documentation generation process.

©IJRASET: All Rights are Reserved

B. Graphical User Interface (GUI)

1) The GUI component built using EasyGUI provides a user-friendly interface for inputting query type and code snippet of the stored procedure.

2) Users select the query type (e.g., "Procedure" or "View") and input the code snippet into the GUI.

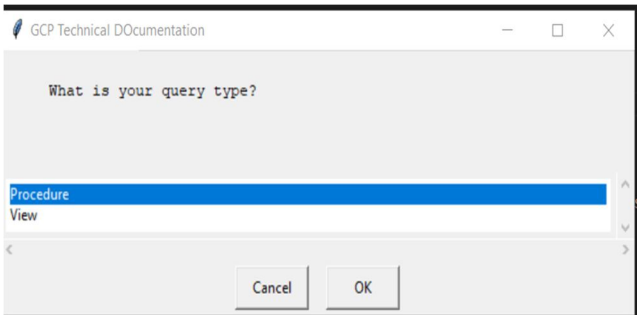


Fig 1.1 GUI Query Type

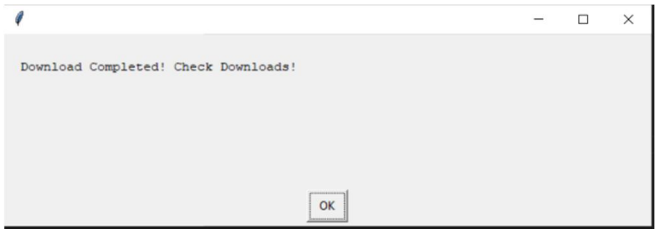


Fig 1.2 GUI Successful Download

C. Input Processing

1) The user-provided input from the GUI, including the query type and code snippet, is processed by the script.

2) Preprocessing modules handle input validation, formatting, and tokenization to prepare the code snippet for further processing.

D. Text Generation using Language Models

1) The preprocessed code snippet, along with a prompt specifying the documentation requirements, is passed to the TextGenerationModel provided by Vertex AI.

2) The language model generates natural language text describing various aspects of the stored procedure based on the input provided.

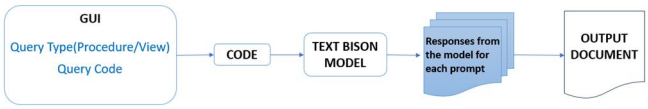


Fig 1.3 Text Bison Model

E. Documentation Structure and Formatting

1) Generated text is parsed and structured into a standardized format suitable for technical documentation.

2) Techniques such as regular expressions and string manipulation are employed to extract key information and organize it into sections such as procedure overview, column mappings, temporary table descriptions, etc.

F. Output Presentation

1) The formatted documentation is presented to the user in a user-friendly format, such as a Word document or PDF.

2) Customization options allow users to select specific sections or details to include in the documentation.

3) Visual elements such as tables, code snippets, and annotations enhance comprehension and clarity.

G. Validation and Quality Assurance

1) Validation mechanisms ensure the accuracy and reliability of the generated documentation.
2) Generated documentation is compared with manually curated references or subjected to peer reviews to verify correctness.
3) Discrepancies or inconsistencies are flagged for further investigation and refinement.

H. Feedback and Iteration

1) Users are encouraged to provide feedback on the generated documentation, facilitating continuous improvement.
2) Feedback is used to refine the text generation model and documentation generation pipeline, enhancing performance and usability.

III. EVALUATION AND VALIDATION

A. Evaluation Metrics

To assess the effectiveness and accuracy of the automated documentation generation process, we propose the following evaluation metrics:

1) *Documentation Completeness:* Evaluate the generated documentation to ensure that it covers all relevant aspects of the BigQuery stored procedure, including procedure overview, column mappings, temporary table descriptions, etc.
2) *Semantic Coherence:* Assess the semantic coherence and clarity of the generated documentation by examining the logical flow of information and the coherence of language used.
3) *Technical Accuracy:* Validate the technical accuracy of the generated documentation by comparing it with manually curated references and expert knowledge. Ensure that the documentation accurately represents the logic, structure, and functionality of the stored procedure.
4) *User Satisfaction:* Solicit feedback from users to gauge their satisfaction with the generated documentation. Collect qualitative feedback on usability, readability, and overall usefulness of the documentation.

B. Validation Process

The validation process involves the following steps:

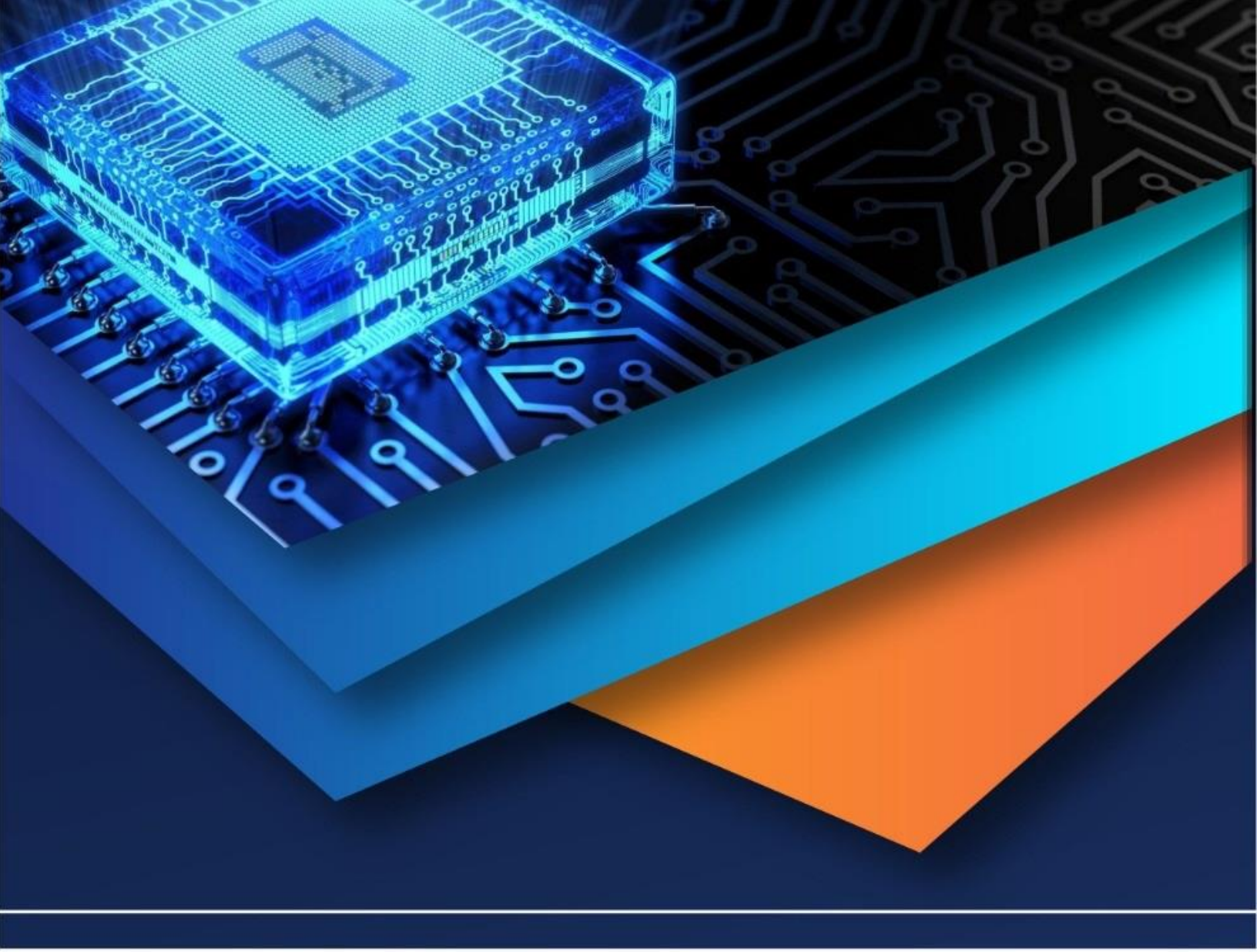
1) *Manual Review:* A team of domain experts manually reviews a sample of generated documentation to assess its completeness, coherence, and technical accuracy.
2) *Comparison with Gold Standard:* Generated documentation is compared with manually curated references or existing documentation considered as the gold standard. Discrepancies or inconsistencies are identified and analyzed.
3) *User Feedback:* Users are invited to provide feedback on the generated documentation through surveys or interviews. Feedback is analyzed to identify areas for improvement and refine the documentation generation process.

C. Results and Analysis

The results of the evaluation and validation process will be analyzed to identify strengths, weaknesses, and opportunities for improvement in the automated documentation generation process. Insights gained from the evaluation will inform iterative refinements to the text generation model, preprocessing pipeline, and user interface to enhance the overall quality and usability of the generated documentation.

REFERENCES

[1] Gao, S., Zhao, L., Lu, H., & Wang, H. (2020). Automated Generation of Technical Documentation for Software Projects: A Systematic Literature Review. IEEE Access, 8, 166376-166392.
[2] Pandey, R., & Rathore, N. S. (2021). A Review on Automated Documentation Generation Tools. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 7(2), 48-53.
[3] Vasilescu, B., Filkov, V., & Serebrenik, A. (2014). StackOverflow and GitHub: Associations between Software Development and Crowdsourced Knowledge. In Proceedings of the 11th Working Conference on Mining Software Repositories (MSR '14), pp. 146-149.
[4] Alhelbawy, A., Ihmouda, M., Kharbutli, M., & Pollock, L. (2019). Unsupervised Extraction of Domain-specific Keywords from Text. In Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP '19), pp. 203-212.
[5] Krishnamurthy, V., Duan, R., & Venkatasubramanian, K. (2021). Generative Adversarial Networks for Automatic API Documentation Generation. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '21), pp. 1411-1415.
[6] Sallam, M. A., Yousef, A. E. M., & Khamis, N. E. M. (2020). Automated Code Documentation Generation: A Survey and Future Directions. International Journal of Advanced Computer Science and Applications (IJACSA), 11(1), 260-269.
[7] McMillan, C., Grechanik, M., & Poshyvanyk, D. (2012). What's There and What's Not: Thoughts on API Documentation. In Proceedings of the 34th International Conference on Software Engineering (ICSE '12), pp. 9-19.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)