www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 10, October 2025, pp : 504-506

e-ISSN : 2583-1062

Impact Factor : 7.001

SMART PROMPT DEBUGGER: AI AT WORK OPTIMIZING PROMPT ENGINEERING

Yash V. Bhonde[1], R.S. Durge[2]

[1]B.E. (CSE) student, Dr. Rajendra Gode Institute of Technology & Research, Amravati, Maharashtra, India, Department of Computer Science & Engineering, India.

[2]Guide, Dr. Rajendra Gode Institute of Technology & Research, Amravati, Maharashtra, India, Department of Computer Science & Engineering, India.

DOI: https://www.doi.org/10.58257/IJPREMS44223

ABSTRACT

This system offers a workflow for prompt optimization assisted by AI, aimed at enhancing user inputs for large language models. Users submit unrefined prompts that the system examines for any ambiguity, lack of clarity, or absent context. Utilizing GPT API integration, enhanced prompt recommendations are produced with more precise wording, increased context, and organized guidelines. A comparison interface shows both the original and enhanced prompts, allowing users to assess and choose the optimal version, which is subsequently stored for future reference and ongoing learning. The system utilizes a frontend constructed with HTML, CSS, and JavaScript (or React), a backend powered by Flask or Node.js, and a database like SQLite or Firebase for monitoring prompt history and analytics. Data security is guaranteed via encryption both during storage and transmission. This method improves the quality of prompts, encourages user learning, and simplifies engagement with AI models.

1. INTRODUCTION

Successful engagement with AI language models relies significantly on the quality of user prompts. Unrefined prompts are frequently unclear, partial, or indistinct, resulting in less than ideal reactions from the AI. To tackle this issue, the suggested system offers an AI-driven prompt enhancement procedure that assists users in improving their inputs. The system assesses the submitted prompts for issues like ambiguity, redundancy, or lack of context, and creates enhanced variations through GPT API integration. Users are able to compare initial and improved prompts, choose the best option, and store it for later use, building a collection of enhanced prompts that fosters ongoing learning. The architecture of the system integrates a responsive frontend (HTML, CSS, JavaScript/React), a powerful backend (Flask or Node.js), AI-based assessment, and a database (SQLite or Firebase) to guarantee effective processing, timely tracking, and secure data handling. This method improves prompt quality, user education, and the overall experience of interacting with AI

2. WORKING & METHODOLOGY

• **User Input:** Users provide unrefined prompts that might be unclear or lacking detail.

• **Prompt Evaluation:** System detects issues such as vagueness, repetition, or lacking context.

• **Enhancement Recommendations:** GPT API creates enhanced prompts featuring more precise wording, additional context, or sequential guidance.

• **Comparison View:** The original prompts and the enhanced prompts are displayed next to each other for assessment and education.

• **User Choice:** Users may approve, decline, or modify prompts; approved prompts are stored for future reference and ongoing learning.

Technical Methodology:

• **Frontend**: HTML, CSS, JavaScript (React optional).

• **Backend:** Flask or Node.js for handling processing and managing sessions.

• **AI Incorporation:** GPT APIs for instantaneous prompt enhancement**.**

• **Database:** SQLite/Firebase for storing prompt history and analytics.

• **Security:** Encrypting data both in storage and during transmission to ensure privacy.

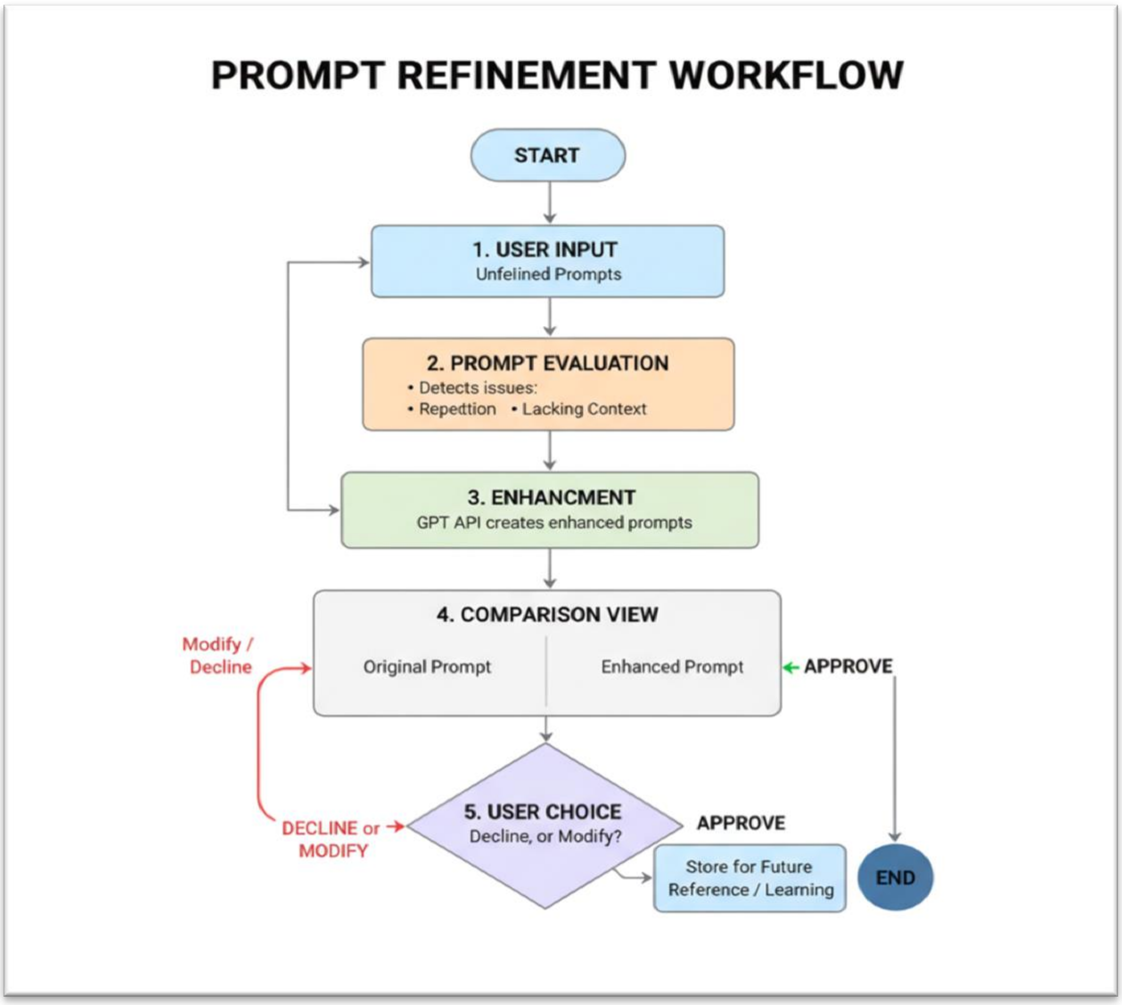


Fig.1 Prompt Refinement Workflow

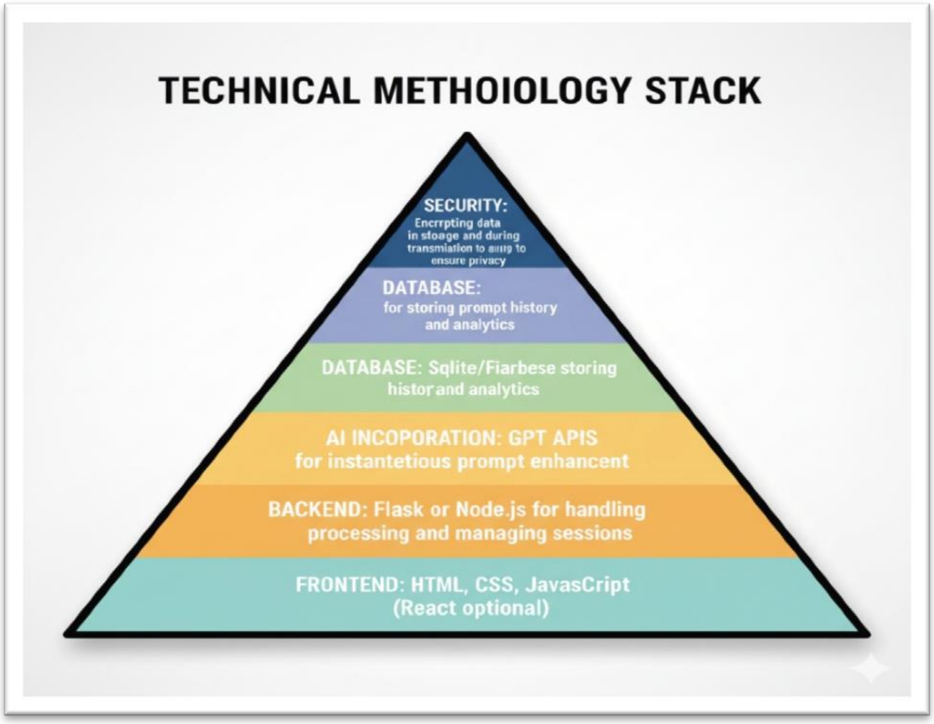
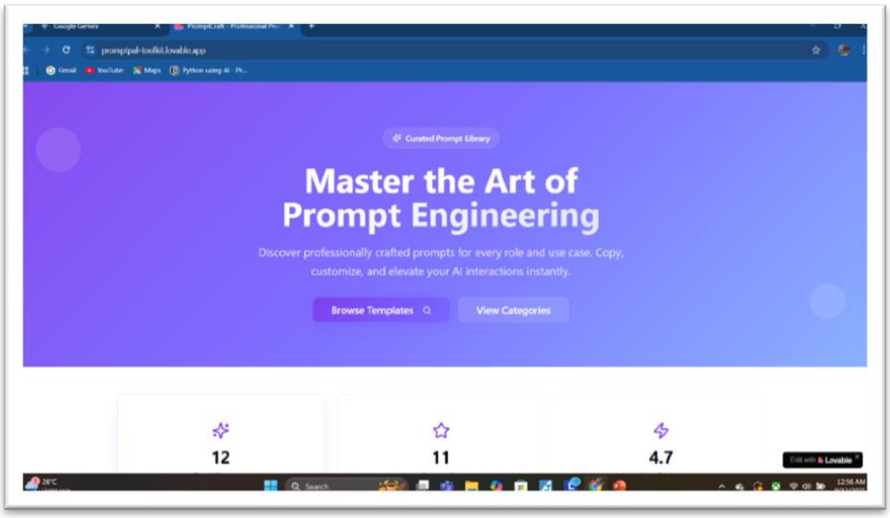

TECHNICAL METHOIOLOGY STACK
SECURITY:
Encrrpting data in sfoage and during transmiation to aiip to ensure privacy
DATABASE:
for storing prompt history and analytics
DATABASE: Sqlite/Fiarbese storing histor and analytics
AI INCOPORATION: GPT APIS for instantetious prompt enhancent
BACKEND: Flask or Node.js for handling processing and managing sessions
FRONTEND: HTML, CSS, JavasCript (React optional)

Fig. 2 Technical Methodology

3. RESULTS AND DISCUSSION

The system successfully enhanced user prompts by detecting ambiguities, redundancies, and lacking context. Suggestions generated by the GPT API improved clarity, specificity, and focus on tasks. The comparison interface allowed users to grasp prompt enhancements, while saved prompts facilitated ongoing learning. The integration of backend and frontend offered a responsive interface, while the database guaranteed secure and swift storage. In general, the system enhanced the quality of prompts, minimized trial-and-error, and improved the relevance of responses generated by AI.

❖ **Implementation :** web page based on this paper
https://promptpal-toolkit.lovable.app/


Master the Art of Prompt Engineering

Discover professionally crafted prompts for every role and use case. Copy, customize, and elevate your AI interactions instantly.

Fig.3 prompt pal toolkit

4. CONCLUSION

The prompt optimization system supported by AI significantly improves the quality of user inputs for language models. Through the examination of raw prompts for vagueness and absent context, creation of enhanced suggestions via GPT API integration, and offering a comparative interface, the system allows users to improve and choose the best prompts. Storing approved prompts establishes a collection for ongoing learning and future use. The combination of a lively frontend, strong backend, and secure database guarantees effective performance, quick tracking, and data protection. This method enhances AI engagement, fosters a deeper comprehension of efficient prompt creation, and simplifies the process of receiving precise and pertinent AI replies

ACKNOWLEDGEMENT

The author is deeply grateful to Dr. R. S. Durge for his invaluable advice and mentoring, and acknowledges the academic support provided by the Department of Computer Science & Engineering at Dr. Rajendra Gode Institute of Technology & Research, Amravati, for this study.

5. REFERENCES

[1] OpenAI Documentation (2025). Prompt Engineering and API Usage Guide. OpenAI Technical Resources. Retrieved from https://platform.openai.com/docs.
[2] Smith, J. (2024). Prompt Engineering Techniques. AI Journal, Vol. 12(3), pp. 45–60.
[3] Johnson, M. (2023). Improving NLP Outputs with Prompt Debugging. IEEE Transactions on Computational Intelligence and AI in Industry, Vol. 9(2), pp. 112 124.
[4] Additional Online Resources and Case Studies (2022–2025).