IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 1, January 2024

Upgrading ChatGPT: Resolving ChatGPT Server Load with Affordable High-Performance Model

Mangala Malkar[1], Pratik Baviskar[2], Abhijeet Gaikar[3], Om Kinage[4], Prathamesh Kumbhar[5]

Head of Department, Department of Computer Engineering[1]

Students, Department of Computer Engineering[2,3,4,5]

Pimpri Chinchwad Polytechnic, Pune, India'

Abstract*: We investigate the creation of ChatGPT templates to solve the problems caused by the high server load of the original system. The main objective is to ensure that users have continuous access, especially during peak usage periods. Thanks to the various applications of ChatGPT, including question answering and content creation, our project aims to create a reliable plugin system that can be managed while the original model is heavy.The importance of this project is to contribute to the democratization of AI and to expand access to advanced language models to a larger community of users. By solving server load issues, ChatGPT clone improves overall accessibility and user experience, and provides AI chats for users. This research is aligned with the broader goal of improving the generalizability and usability of AI technologies by promoting accessibility and reliability.*

Keywords: Ensures uninterrupted access& affordable GPT models

I. INTRODUCTION

The primary motivation behind developing a ChatGPT clone is to address the issue of server load and ensure continuous, reliable access for users. In situations where the original ChatGPT experiences high traffic or usage, there may be delays or limited availability. By creating a clone, we aim to mitigate these challenges, maintaining a seamless and responsive experience for users.

This project acknowledges the significance of ChatGPT as a valuable tool for various applications, including answering questions, generating content, and assisting users. A clone can serve as a reliable backup, stepping in when the primary system is overloaded, thus enhancing overall accessibility and usability. Moreover, this initiative aligns with the broader goal of democratizing AI, making advanced language models accessible to a larger audience. Through a ChatGPT clone, we aim to promote widespread availability and reliability, ensuring that users can harness the power of AI-driven conversations without interruption.

II. METHODOLOGY

1. User interface development: The project begins with the development of a responsive and visually appealing React user interface. The user interface is thoroughly tested to ensure an exceptional user experience.

2. Backend infrastructure: Node.js backend is set up for user authentication, API key management and chat session management. The backend is robust and scalable for future growth.

3. AI Integration: OpenAI's GPT-3.5 Turbo is seamlessly integrated into the app, allowing the chat to provide intelligent and contextual responses based on user input.

4. Chat history: a versatile chat history system is introduced to allow users to easily view, search and manage their chat history.

5. Scalability and Optimization: The application is optimized to ensure that it can handle increasing numbers of users while maintaining responsive and reliable performance. With this project, we aim to provide a cutting-edge chat that not only offers the convenience of AI-powered communication, but also prioritizes user experience, security and scalability. This program sets new standards for online communication by seamlessly integrating AI capabilities into everyday conversations, improving both personal and professional interactions.

DOI: 10.48175/568

III. LITERATURE REVIEW

The development of a ChatGPT clone presents two fundamental approaches: employing dataset-driven model training or integrating existing APIs. Each approach brings distinct advantages and considerations.

A. Model Training vs. API Integration:

The project can undertake the arduous process of training a chatbot model from datasets, ensuring its comprehension of diverse language patterns. Alternatively, leveraging established APIs like OpenAI's provides a shortcut to harness the power of pre-trained models, thereby saving time and enhancing response accuracy through state-of-the-art language understanding capabilities. We'll be using the API in our project.

B. API Key Integration (OpenAI API Key):

The integration of the OpenAI API key is pivotal to the project's success. This API key facilitates direct interaction with the sophisticated language model and enables the AI to comprehend user inputs and formulate contextually coherent responses. This connection between the application and the OpenAI API empowers the chatbot clone to engage users in natural conversations, mirroring the proficiency of the original ChatGPT.

C. Node.js & Express.js Backend:

Node.js serves as the core backend technology, orchestrating server-side logic and managing API requests. Express.js, employed in conjunction with Node.js, streamlines the development of the web application's routing and middleware, simplifying the creation of robust APIs. This combination ensures efficient handling of user interactions and AI chatbot integrations, facilitating real-time communication and enabling a responsive, data-driven chat application on the client-side. Together, Node.js and Express.js empower the project to deliver a seamless and dynamic user experience.

D. User Interface with React.js & CSS:

React is the cornerstone of the frontend, enabling the creation of a dynamic user interface and facilitating seamless user interactions. React's component-based architecture simplifies the organization and maintenance of the application's UI elements, enhancing code reusability and modularity. CSS is employed to style the user interface, providing visual appeal, and ensuring a consistent and visually engaging design. Together, React and CSS synergize to deliver an intuitive, visually appealing, and responsive chat application that effectively engages users while ensuring a seamless user experience.

IV. RESULTS AND DISCUSSION

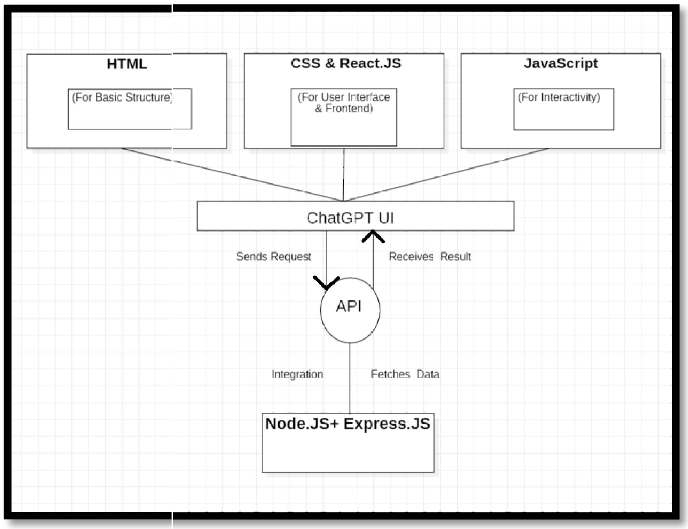


Fig. 1. Working of ChatGPT

IJARSCT

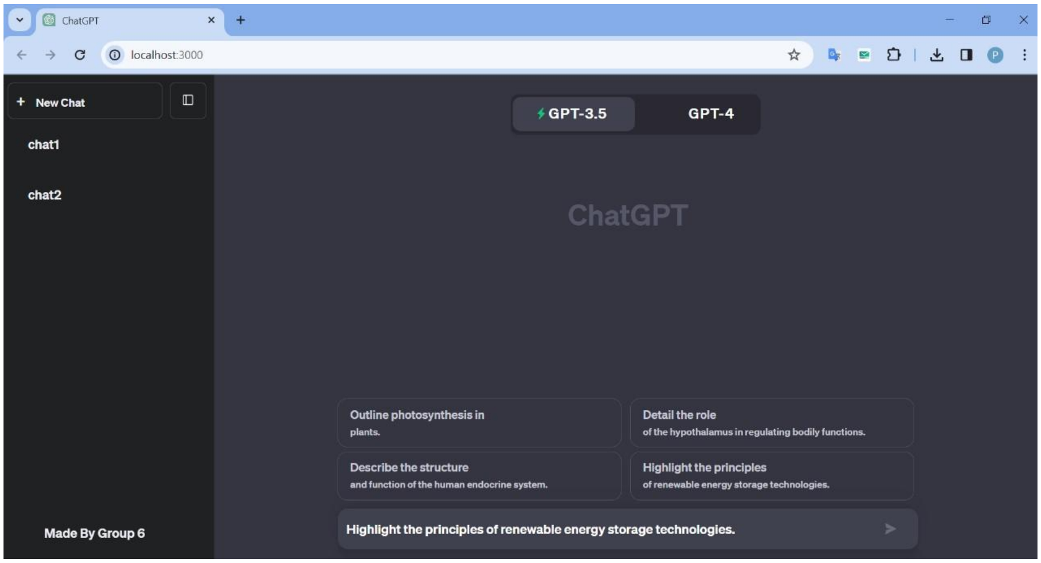
ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 1, January 2024



Outline photosynthesis in plants.

Detail the role of the hypothalamus in regulating bodily functions.

Describe the structure and function of the human endocrine system.

Highlight the principles of renewable energy storage technologies.

Highlight the principles of renewable energy storage technologies.

Fig. 2. Frontend of ChatGPT

This project uses OpenAI and GPT-3.5 Turbo model and GPT-4 model to implement chat application. Users can use the chat application while the original ChatGPT is under server load. Also, you can access GPT-4 for free at first, and the price will decrease over time. Users can interact with AI assistants through a user-friendly interface, start conversations and seamlessly switch between different chat sessions. The app dynamically displays user input and AI-assisted responses to deliver engaging and interactive experiences. This application also includes various styles to enhance your presentation, creating an elegant and user-friendly experience. Sensitive information, such as OpenAI API keys, is handled securely using the dotenv package. The API integration is solid, handles asynchronous requests well, and implements error handling mechanisms to handle potential problems when working with the OpenAI API.

The interpretation of the results, the application of the discussion, and its success depend on the effective integration of OpenAI with the GPT-3.5 Turbo model and the GPT-4 model. Dynamic message display, chat history scheduling and smooth communication contribute to a good user experience. Image improves image quality and makes your app more attractive to the user. By combining these results with the existing literature, this project joins the growing trend of using high-level language models for interactive applications. Known for its natural language generation and recognition capabilities, OpenAI and its GPT-3.5 Turbo and GPT-4 models are powerful tools for creating communication interfaces. The research problem is to create a functional chat application using OpenAI and API #039;. The results show that we achieved this goal by focusing on user experience, security and effective communication with the AI model. In a broader context, this project is linked to the ongoing exploration of natural language processing and AI interaction. The articles in this section highlight the challenges and opportunities for creating simple communication between humans and AI models.

V. CONCLUSION

This project encompasses the development of a user-friendly chat application with AI integration, allowing users to engage in natural conversations. The scope includes building a React frontend, Node.js backend, secure authentication, API key management, chat history, and scalability for a seamless user experience. The primary objective is to deliver a robust, AI-powered chat platform that enhances user interactions.

Input: It can be a single sentence, a paragraph, or even a series of prompts or questions.

Output: This is the response generated by the model based on the provided input. It can be a single sentence, a paragraph, or even longer.

VI. ACKNOWLEDGEMENTS

We want to express our gratitude to Professor M.S. Malkar, Head of the Department, for her invaluable guidance and constant support in the research project. Her deep experience and knowledge were instrumental to the project and its successful completion. We would also like to thank Techtrix Solutions for their great support, who sponsored the project and its results. Also, I want to express my sincere gratitude to my peers who helped make the project possible. Their support and contributions greatly enriched our research efforts.

REFERENCES/APPENDICES

[1] Open AI API Key Documentation. https://platform.openai.com/docs/quickstart
[2]Node.JS https://nodejs.org/en/docs
[3]React.JS https://react.dev/
[4]Express JS https://expressjs.com/