Backend Developer Home Assignment, Prompt Security

LLM gateway

Objective

Develop an LLM gateway service designed to safeguard sensitive data by sanitizing its transfer to/from an LLM.

The gateway will serve as an HTTP proxy between the client and the LLM: sending the safe prompt to the LLM and afterwards returning its response to the client.

Requirements

1. Use TypeScript throughout the project. Enforce strict typing, avoid using any, and demonstrate good type definitions.
2. Gateway API - provide a RESTful API endpoint to receive and process prompts:
   - Implement robust API key-based authentication to ensure secure access.
   - Sanitize incoming prompts containing email addresses before sending it to the LLM; for example, a prompt containing john@gmail.com should change to xxx@my.email.
   - Integrate efficiently with a local LLM of your choice (e.g., Jan, Llama.cpp, llama or any other of your choice).
   - Inspect returned responses from the LLM to sanitize email addresses in the same manner described above.
   - Return to the client a response that includes the sanitized response, along with a list of email addresses that were sanitized from the prompt/response.

Bonus

Gateway cache - for better performance of your gateway, implement a caching mechanism.

- The cache should store frequently used prompts, responses, and their sanitized entities. This approach avoids redundant sanitization of commonly processed text and improves overall efficiency.

Submission Guidelines

To submit your completed assignment, please provide us with a single folder/github repository containing -

- The service
- Instructions for using the service
- Explanation for your technical choices - which libraries you chose to use and why? which databases you chose and why?
- List of limitations that the service have (if any)
- List of possible features/requirements that the services needs in order to make it production ready
- List of performance improvement ideas to support large scale