

1 - Chrome Extension

- I created a Chrome extension with React, Next.js, and TypeScript to censor hateful tweets on Twitter. Twitter's dynamic rendering of posts presented the primary technical obstacle. While users scrolled, posts were continuously taken out and put back into the DOM with varying IDs, causing challenges in consistently monitoring and organizing tweets.
-
- In order to solve this issue, I set up a VectorDB system and a Redis server to store all tweets instantly. This enabled quicker semantic search and effective categorization of tweets. If the Twitter post's embedding was located in the VectorDB, it was promptly blurred.
-
- In order to guarantee efficiency and the ability to grow, I enhanced the backend by utilizing FastAPI and NLP algorithms. Using TypeScript aided in the preservation of code quality and early detection of possible errors during the development phase. Utilizing React for the front end in conjunction with a strong back end structure enabled the creation of a flexible and expandable single-page application capable of managing the constantly changing content loading on Twitter.

2 -CodeMonk SWE

- While working as a Software Engineer at Codemonk, I obtained important skills in creating scalable microservices and APIs with Java and Spring Boot. I set up a Redis server to cache commonly used data, improving system performance and scalability by decreasing redundant processing and database queries. This method was especially successful in handling heavy traffic while maintaining performance.
-
- I also applied horizontal scaling by integrating additional worker servers to the backend API, enabling effective management of heightened requests in the queue. This approach guaranteed that our system could handle increased levels of traffic while still operating at peak performance. By utilizing these methods, I helped construct a strong and adaptable backend system that could easily blend with intricate backend structures.