

Backend Engineering Launchpad – Case Study

[AI + Backend] WebGPT

Author: Airtribe

Background & Problem Statement

In the digital age, the vast amount of information available on websites presents both an opportunity and a challenge. While the internet is a rich source of knowledge, efficiently navigating and extracting specific information from web pages can be daunting for users. This complexity is compounded by the diverse structure and presentation styles of websites. Traditional search engines provide a broad solution for information retrieval, but they often lack the capability to offer precise, context-specific answers from individual websites. There is a growing need for more intuitive and interactive methods to engage with and understand web content.

Objectives

1. **System Design:** Create an efficient backend system for crawling and loading website content.
2. **Robustness and Accuracy:** Develop a system capable of handling high volumes of queries and delivering accurate results.
3. **Scalability:** Ensure the system can scale to accommodate increasing volumes of website content and user queries.

4. **Timely Responses:** Provide accurate and prompt responses to user queries.

Scope

The project's scope is centered on backend development, including system architecture design, algorithm development, and backend implementation. It excludes front-end user interface development, though a minimal front-end addition is desirable.

Expected Outcomes

1. A backend system capable of processing and extracting website content for a given URL.
2. An algorithm or integration to analyze website content and respond to user queries based on this content.
3. Accurate and prompt responses to user queries.

Assessment Criteria

1. **Functionality:** Does the system work as intended? Does it meet all the requirements stated above?
2. **Code Quality:** Is your code clean, organized, well-commented, and following best practices?
3. **Design and Structure:** Is the system well-designed? Does it demonstrate a good understanding of system design principles and patterns?

4. **Documentation:** Is your report comprehensive and clear?
Does it effectively explain the choices made and how to use the project?
5. **Presentation:** Do you effectively demonstrate and explain your system and the decisions made during its development?

Deliverables

1. The final, functional product.
2. README file outlining how to use the system, API documentation, the design decisions and other necessary information.
3. Public link of the Github repository
4. Explainer video demonstrating your project work