Problem Statement

Users waste time scanning through web pages to find the information that actually answers their search intent. Search engines stop at the results page — once a user clicks a link, they're left digging through content alone.

Target Audience

Worsin AI is designed for anyone who frequently researches, compares, or fact-checks online. This includes students, researchers, professionals, shoppers comparing products, and knowledge workers who want faster, more focused browsing. The extension aims to reduce cognitive load, help users stay oriented, and make web exploration more efficient — no matter the domain.

Use of Generative AI

Worsin AI applies Generative AI in two ways:

1. It generates short, clear summaries of the most relevant sections on a webpage so users can grasp key points at a glance.
2. It uses embedding models (like OpenAI's text-embedding-3-small) to compare user queries with page content and identify matching sections, enabling precise highlighting and easy filtering across multiple pages.

Core Idea & Approach

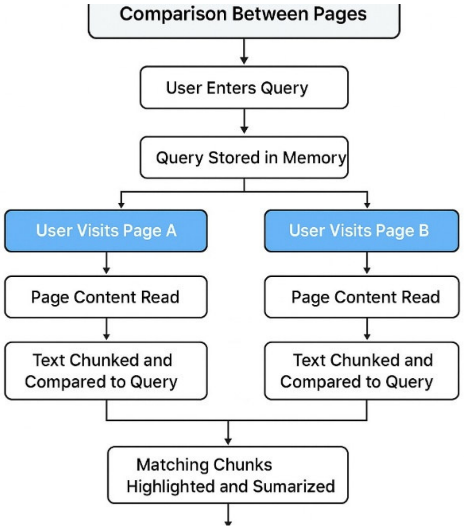
The core idea is to turn passive browsing into guided exploration. Worsin AI stays with the user across pages, highlights the parts of a webpage that closely match their query, and presents easy-to-read summaries in a sidebar.

Our approach is practical for a 48-hour build:

1. Use Chrome extension APIs to read page content, inject highlights, and create the sidebar.
2. Use OpenAI's GPT API for on-demand summarization of relevant text chunks.
3. Use OpenAI's text-embedding API to compute similarity between user queries and text on the page.
4. Store matched chunks and their summaries locally (via local storage or in-memory) so users can compare relevant parts across multiple pages.
5. No heavy backend — all logic stays client-side for speed, simplicity, and privacy.

Workflow and Architecture

- User enters query in the extension.
- Extension stores the query.
- On every page, it scans visible content, breaks it into chunks (e.g. paragraphs, headings).
- For each chunk, compute similarity to the query using embeddings.
- Highlight matching sections and generate summaries.
- Display both current page highlights and top matched sections from previous pages in the sidebar.



Tech Stack:

1. Chrome Extension APIs for DOM reading & UI
2. OpenAI APIs (embeddings + GPT)
3. Local storage for query + matches

Feasibility and Execution

Our plan is realistic for the hackathon:

- We're using tested APIs and tools that integrate easily (Chrome APIs + OpenAI APIs).
- The scope is focused: query → match → highlight → summarize → display.
- No backend setup or complex infrastructure — everything runs inside the browser, allowing us to build, test, and demo within 48 hours.

Scalability and Impact

Worsin AI can scale by adding:

- Browser support beyond Chrome (e.g. Edge, Firefox)
- More advanced query memory for multi-session use
- Paid plans (e.g. query-limit subscription)

It has strong potential to improve how millions of users interact with information online, making web research faster, clearer, and smarter.

Conclusion

Worsin AI makes browsing more focused, efficient, and user-friendly — and it’s practical to build and demo within the hackathon.