

Team.2학년 아이디어 노트

1. 팀 정보

- 팀명: Team.2학년
- 팀원 및 역할

이름	학과	역할
김민섭(팀장)	컴퓨터공학과	프론트엔드(메인), 백엔드(보조), 인공지능(메인)
김정민	컴퓨터공학과	PM
곽민혁	컴퓨터공학과	프론트엔드(메인)
손호영	컴퓨터공학과	백엔드(메인), 인공지능(보조)

2. 탄소중립을 주제로 한 앱 아이디어 자유롭게 작성

- 나만의 개인비서가 검색 결과를 요약 및 필터링해주는 Edge 확장 프로그램
- 아이디어 핵심 키워드
검색 개선, 필터링, 탄소배출량 감소, sLM

3. 문제 정의

3.1. 어떤 탄소 중립 관련 문제를 해결하고자 하나요?

인터넷 검색 시 사용자는 불필요한 광고, 정보가 부실한 블로그, 클릭 유도형 콘텐츠에 자주 노출됩니다.

이러한 사이트를 방문하고 로딩하는 과정에서 불필요한 트래픽과 서버 자원 소비가 발생하며, 이는 곧 데이터 센터의 에너지 사용 증가로 이어집니다.

검색 효율이 낮아질수록 사용자의 검색 시간이 길어지고, 반복적인 데이터 요청이 증가하여 탄소 배출량 증가의 원인이 됩니다.

3.2. 이 문제를 해결하고자 하는 개인적인 동기나 배경, 이유는 무엇인가요?

우리는 일상 속에서 자주 검색을 이용합니다. 하지만 양질의 정보에 효율적으로 도달하지 못하는 경우가 많다는 불편을 느꼈습니다.

특히 학생이나 직장인들이 여러 개의 광고 페이지를 거쳐야 하거나, 원하는 정보를 찾지 못하여 반복 검색을 하는 상황이 매우 비효율적이라고 느껴졌습니다.

이러한 **디지털 낭비(digital inefficiency)**는 단순한 시간 낭비를 넘어 눈에 보이지 않는 탄소 배출로 이어질 수 있다는 점에서 이를 개선하고자 합니다.

3.3. 이 아이디어가 어떻게 탄소중립에 기여하나요?

EnhanceSearch는 직접적으로 탄소를 줄이는 서비스는 아니지만, **디지털 정보 탐색 방식 자체를 전환함으로써 간접적으로 탄소중립에 기여합니다.** 다음과 같은 방식으로 탄소 절감 효과를 기대할 수 있습니다:

- **불필요한 페이지 접근 최소화** → 사용자의 클릭 수 및 데이터 요청 감소 → **네트워크 트래픽과 서버 로딩 감소**
- **빠른 검색 결과 도달** → 검색 반복 횟수 및 체류 시간 단축 → **에너지 소비 절감**
- **사용자 성향 기반 최적화** → 유의미한 정보 중심의 탐색 유도 → **검색 효율 향상**
- *경량 검색 모델(sLM)**을 활용하여, 요즘 폭증하는 **LLM(GPT 등)** 기반 질의 응답 시스템보다 훨씬 적은 연산 자원으로 동일한 탐색 목적을 달성 가능
- **EnhanceSearch는 대부분의 연산을 사용자 개인 컴퓨터에서 처리함으로써, 중앙 집중형 데이터센터의 부하를 줄이고 전력 소비를 분산시킵니다.**

이는 결과적으로 **클라우드 의존도를 낮추고, 탄소 집약적인 대규모 AI 인프라의 사용량을 완화하는 데** 기여합니다.

4. 사용 대상자

- **대학생 및 연구자:** 연구 시 고품질 정보에 빠르게 도달하고자 하는 사용자
- **일반 직장인:** 효율적인 정보 탐색을 통해 업무 생산성을 높이려는 사용자
- **환경에 관심 있는 사용자:** 디지털 소비 또한 탄소중립에 영향을 미친다는 점을 인식하고 있는 사용자
- **기술 친화적인 인터넷 사용자:** 브라우저 확장 프로그램 사용에 익숙한 사용자층

5. 기술 및 서비스 구성 요약

EnhanceSearch는 사용자의 검색 경험을 향상시키기 위해 브라우저 확장 프로그램 형태로 제공되는 경량형 검색 보조 도구입니다.

불필요한 저품질 콘텐츠를 필터링하고, 사용자의 피드백을 바탕으로 검색 결과를 개인화하며,

짧고 애매한 검색어를 자동으로 보완함으로써 더 정확하고 효율적인 검색을 가능하게 합니다.

대규모 LLM이 아닌 경량화된 [sLM\(Gemma 2:2b\)](#) 을 사용하여, 서버 부하를 줄이고 개인 디바이스 중심의 처리 구조를 지향합니다.

5.1. 사용 기술 및 도구

구성 요소	기술 스택 및 도구 설명
Frontend	Next.js, TypeScript, Tailwind CSS
Backend	Spring Framework (Java 기반 REST API), MS Azure 환경에서 서버 운영
AI 서버	Python 기반 Gemma 2:2b 모델, FastAPI로 구성된 경량 AI 처리 서버
사용자 데이터	Firebase (사용자 피드백 및 설정 저장)
디자인 도구	Figma (서비스 프로토타이핑 및 UI/UX 설계)

5.2. 전체적인 서비스 흐름 (사용자 여정)

[확장 프로그램 설치] → [검색어 입력] → [불필요한 결과 자동 필터링]
→ [좋아요/싫어요 피드백] → [개인화된 검색 결과 개선]

5.3. 핵심 기능 및 예상 사용자 반응

핵심 기능	사용자 예상 반응
광고성/저품질 페이지 필터링	원하는 정보에 빠르게 도달, 검색 피로도 감소
검색어 보정 (Gemma2 기반)	검색 실패율 감소, 더 나은 결과 도출
피드백 기반 개인화 추천	맞춤형 결과 제공 → 만족도 및 재사용 증가

5.4. 사용자 반응 (심리적/사회적 유도 전략)

- 검색 결과 정제 → 시간 절약과 효율성 체감
- '좋아요' 참여 유도 → 사용자 주도형 검색 경험 제공
- GPT 기반 질문형 검색보다 더 가볍고 정확한 대안 인식
- 개인 기기에서 처리 가능 → 나만의 검색 비서처럼 인식되어 지속 사용 동기 부여