

## Team. TEXT OPERATION

고가은 김진우 최혜윤

TEXT  
CALCULATION

## 배경 및 문제 의식

최근 몇 년 사이, '텍스트힙(Text Hip)'이라는 개념이 등장하며 책 읽기(Reading) 문화에 새로운 변화가 일어났다. '텍스트힙'은 '텍스트(Text)'와 '힙하다(Hip, 멋있다)'의 합성어로, 독서 행위가 멋지고 세련된 자기표현의 방식으로 인식되는 현상을 뜻한다. 특히 디지털 기술에 익숙한 세대일수록, 오히려 아날로그적인 독서 문화로 회귀하는 경향을 보이며, 이는 단순한 취미를 넘어서 사회적 소통과 감각적 취향의 일부로 자리잡고 있다.

반면 쓰기(Writing)의 영역에서는 정반대의 흐름이 이어지고 있다. 생성형 AI 기술이 발달하면서 사람들은 점점 더 많은 글쓰기 과정을 AI에 의존하게 되었다. 다양한 AI 도구가 빠르게 완성된 문장을 제공하면서, 스스로 문장을 구성하거나 표현을 실험하는 일은 줄어들고 있다. 이제 많은 사용자들은 문장을 '쓰기'보다 '선택'하거나 '복사'하는 데 익숙해졌고, 그 결과 언어 감각과 창의적 문장 구성 능력은 약화되고 있다. 쓰기는 더 이상 사고의 결과물이 아니라, '즉시 소비되는 텍스트'로 취급되고 있다.

이러한 문제의식 속에서 우리는 '쓰기에서의 텍스트힙(Text Hip)'이라는 개념을 제안한다. 이는 단순히 정보를 전달하는 것이 아니라, 문장을 구조화하고 실험하면서 자신만의 표현 방식을 만들어가는 행위 자체에 주목한다. 창작자가 언어의 규칙을 직접 설계하고, 감정과 사고를 조합해 자신을 표현하는 과정이야말로 오늘날 가장 힘찬 쓰기 방식이 될 수 있다는 관점이다.

이러한 철학을 바탕으로 우리는 **Text Calculation**이라는 프로젝트를 시작하게 되었다. 이 서비스에서 사용자는 직접 텍스트 공식을 만들고, 그 구조 안에서 창작을 실행할 수 있다. 감정, 대상, 행동 같은 단어들을 수학 연산 기호처럼 조합하여 공식을 만들고, AI는 그 공식을 계산해 문장을 생성한다. 예컨대  $\sqrt{\text{행복}}, \text{감정} + \text{공간}, \partial(\text{사랑})$ 과 같은 표현은 감정을 다층적으로 해석하고 언어화할 수 있는 방식이 된다.

**Text Calculation**은 글쓰기의 주도권을 창작자에게 되돌리는 시도이자, 언어 감각을 훈련할 수 있는 창의적 실험 도구다. 우리는 누군가 대신 써주는 시대를 지나, 이제는 글쓰기의 구조 자체를 사고하고 디자인하는 시대에 진입하고 있다. 이러한 변화 속에서 우리는 '쓰기의 텍스트힙'을 실천할 수 있어야 한다. 즉, 글을 잘 쓰는 것이 중요한 것이 아니라, 어떻게 쓰는지를 설계하고 구성하는 과정이 더욱 주목받아야 한다는 것이다. 이 과정에서 AI는 창작을 대신해주는 존재가 아니라, **인간의 창의적 사고를 계산해주고 확장시켜주는 파트너**로 자리매김해야 한다.

## Text operation 체계

**Text Calculation**은 텍스트를 수학처럼 구조화하여 다루기 위해 다양한 **연산 기호**를 도입했다. 각 연산자는 단어와 문장 사이의 관계를 규정하는 **의미적 기능**을 가지며, 사용자는 이를 활용해 문장을 구조적으로 조합할 수 있다.

- + (결합): 서로 다른 개념의 연결을 통해 새로운 의미를 만듭니다.  
예시: 감정 + 장소 → 감성적 풍경
- (제거): 한 요소를 뺀으로써 \*\*핵심적 의미\*\*를 부각시킵니다.  
예시: 도시 - 소음 → 평화
- × (증폭): 두 요소가 만났을 때 서로의 힘을 키웁니다.  
예시: 자연 × 시간 → 치유
- ÷ (나눔): 의미를 분산하거나 단순화하여 전달력을 높입니다.  
예시: 사랑 ÷ 사람 → 친절, 연민, 애착
- ✓ (본질): 개념의 근원적 속성이나 감각을 추출합니다.  
예시: ✓커피 → 향, 카페인, 온기
- Σ (총적): 반복이나 누적을 통해 전체적인 흐름을 나타냅니다.  
예시: Σ(노력 + 실패) → 성장
- ʃ (흐름): 시간성과 감정의 연속성을 강조하는 표현  
예시: ʃ(여행 + 풍경) → 서사, 회상, 기억
- ∂ (변화): 특정한 순간의 감정 변화나 감각의 전환을 포착합니다.  
예시: ∂슬픔 → 그리움, 눈물, 포기

## 생성방식 (출력과 해석)

사용자가 입력한 연산식에 따라 시스템은 두 가지 출력을 제공한다:

```
{
  "text_output": "operation 결과"
  "output_detailed": "왜 그 문장이 만들어졌는지에 대한 구조적 해석"
}
```

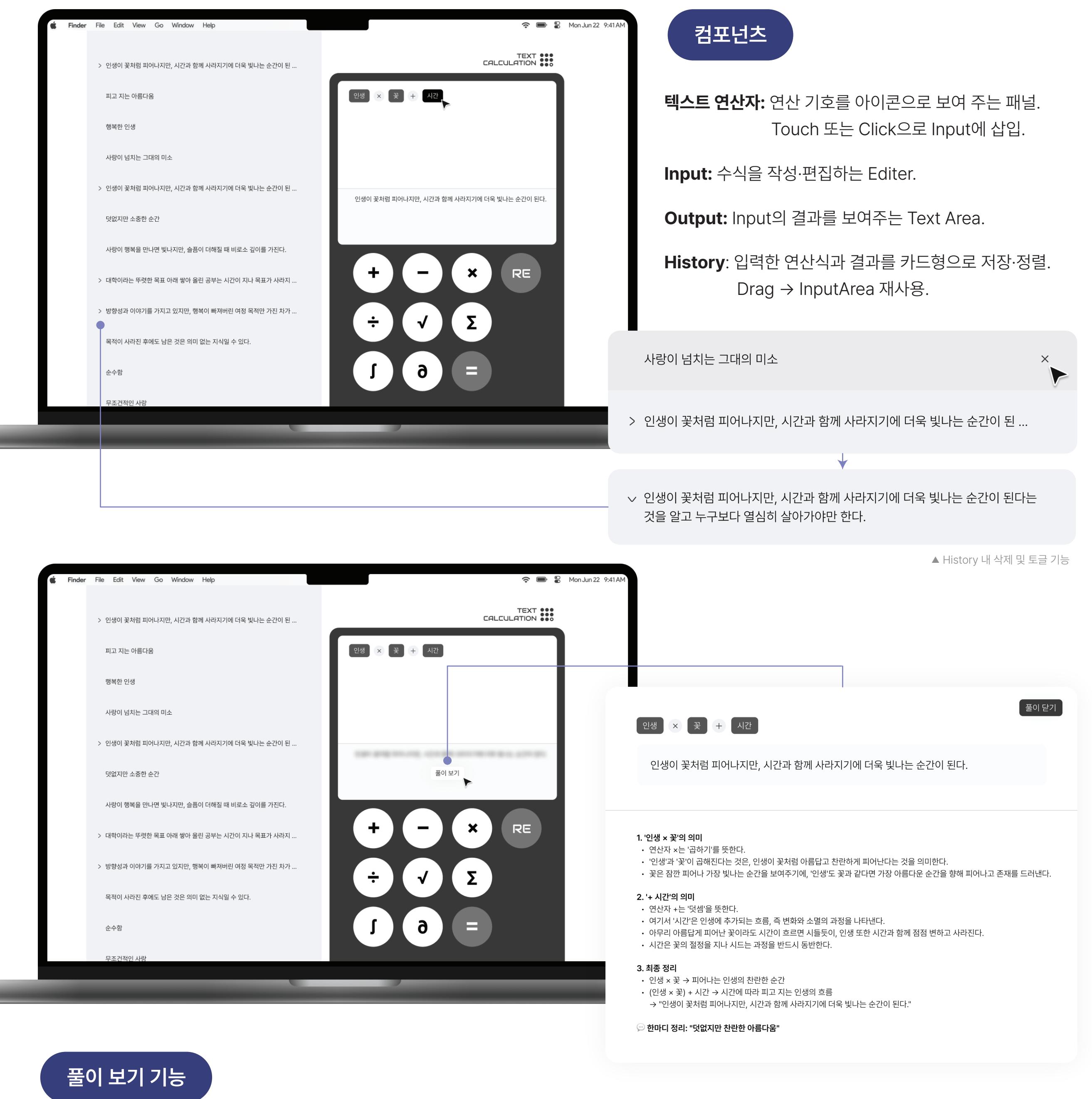
예를 들어,

입력: **✓행복**  
→ text\_output: "사랑이 남았다"  
→ output\_detailed:

✓연산은 개념의 본질을 추출하는 기호이다. '행복'이라는 단어에서 가장 핵심적인 감정이나 구성 요소를 추출했을 때, 사랑, 민족, 미소 등의 정서적 단위로 환원할 수 있다. 여기서 사랑을 선택한 것은, 다양한 행복의 형태 중에서도 가장 보편적이고 핵심적인 감정 단위이기 때문이다.

이는 단순히 결과만을 보여주는 데 그치지 않고, 그 결과가 어떻게 도출되었는지를 함께 제공한다. 이를 통해 사용자는 자신의 의도와 연산 논리를 어떻게 표현으로 이어졌는지를 검토할 수 있으며, 결과에 대한 자기 피드백과 창작 실험을 반복할 수 있는 구조적 기반을 마련하게 된다.

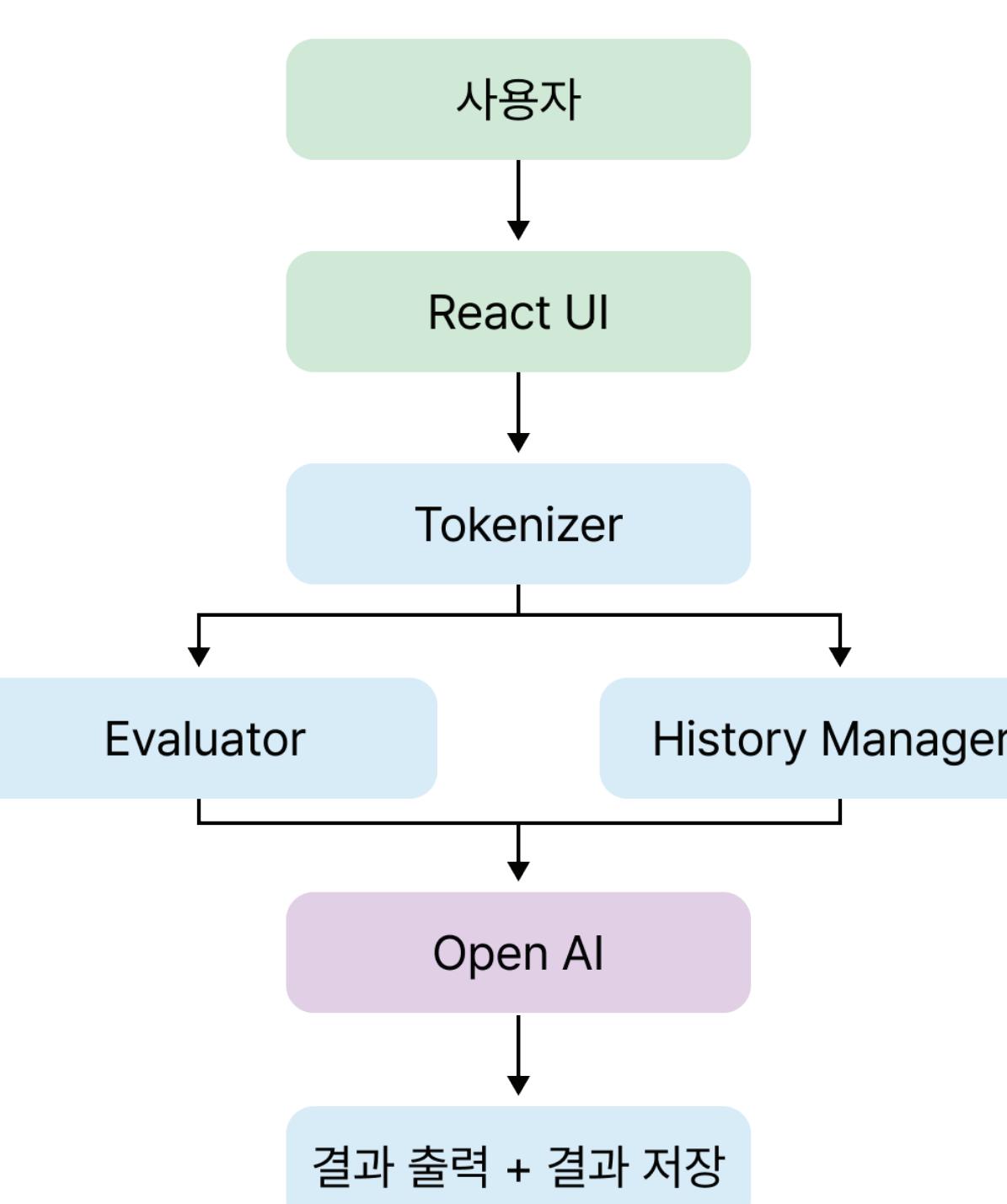
## UI/UX 측면



## 풀이 보기 기능

- 결과 영역 Hover 시 나타나는 '풀이 보기' 버튼 Click → output\_detailed 구조 해석을 Modal 형식으로 표시.
- 각 연산 단계를 단계별로 강조하여, 사용자가 계산 과정을 직관적으로 이해할 수 있도록 지원.

## 사용된 기술



## 기술 스택

- Frontend: React.js, Tailwind CSS
- Backend (optional): OpenAI API (GPT-3.5 / GPT-4)
- Build Tool: Vite
- Deployment: Netlify / Vercel (GitHub 연동)

## 구성 요소

- InputArea: 사용자가 텍스트 입력
- Tokenizer: 입력값을 연산자 기준으로 텍스트/연산자 토큰으로 분리
- Evaluator: 연산 수행(Open AI)
- History Manager: 연산결과 저장 및 재사용(드래그, 드롭 지원)
- Result Box: 결과 표시 및 모달로 상세보기 지원

## 결론

**Text Calculation**은 AI가 글을 대신 써주는 방식이 아니라, 사용자가 직접 창작의 규칙을 설계하고, AI는 그것을 계산하는 구조를 중심에 둔 실험적 글쓰기 도구이다. 단어들을 연산 기호와 함께 조합함으로써, 사용자는 자신의 사고 흐름을 구조화할 수 있고, 그 결과로 생성된 문장과 해석을 통해 반복적인 피드백과 창작 실험을 가능하게 한다. 이로써 사용자는 단순히 결과를 소비하는 것이 아니라, 표현의 원리를 설계하고 조정하는 창작의 주체로 기능하게 된다.

이는 생성형 AI 시대에 글쓰기의 방향성을 재정의하려는 시도이자, '쓰기의 텍스트힙'을 구현하기 위한 첫걸음이다. 이는 생성형 AI 시대에 글쓰기의 방향성을 재정의하려는 시도이자, '쓰기의 텍스트힙'을 구현하기 위한 첫걸음이다. 이는 생성형 AI 시대에 글쓰기의 방향성을 재정의하려는 시도이자, '쓰기의 텍스트힙'을 구현하기 위한 첫걸음이다. 이는 생성형 AI 시대에 글쓰기의 방향성을 재정의하려는 시도이자, '쓰기의 텍스트힙'을 구현하기 위한 첫걸음이다. 이는 생성형 AI 시대에 글쓰기의 방향성을 재정의하려는 시도이자, '쓰기의 텍스트힙'을 구현하기 위한 첫걸음이다.

항목	항목
프로젝트명	Text Calculation
팀명	Text Operation
팀원	고가은, 김진우, 최혜윤
기술 스택	React, GPT API, Zod, Figma
역할 분담	기획·개발 (가은), 개발 (진우), 디자인 (혜윤)