

Emotionary AI

일기 데이터 기반 감정 기록 서비스

5조 : 강혜정 김재혁 임다빈 나상훈 김은혜

Read More ➤



오늘 당신의 마음은 어떤가요?

복잡한 생각들, 누구에게도 하지 못한 이야기들...

Emotionary AI가 당신의 소중한 감정을 보듬어 드릴게요.

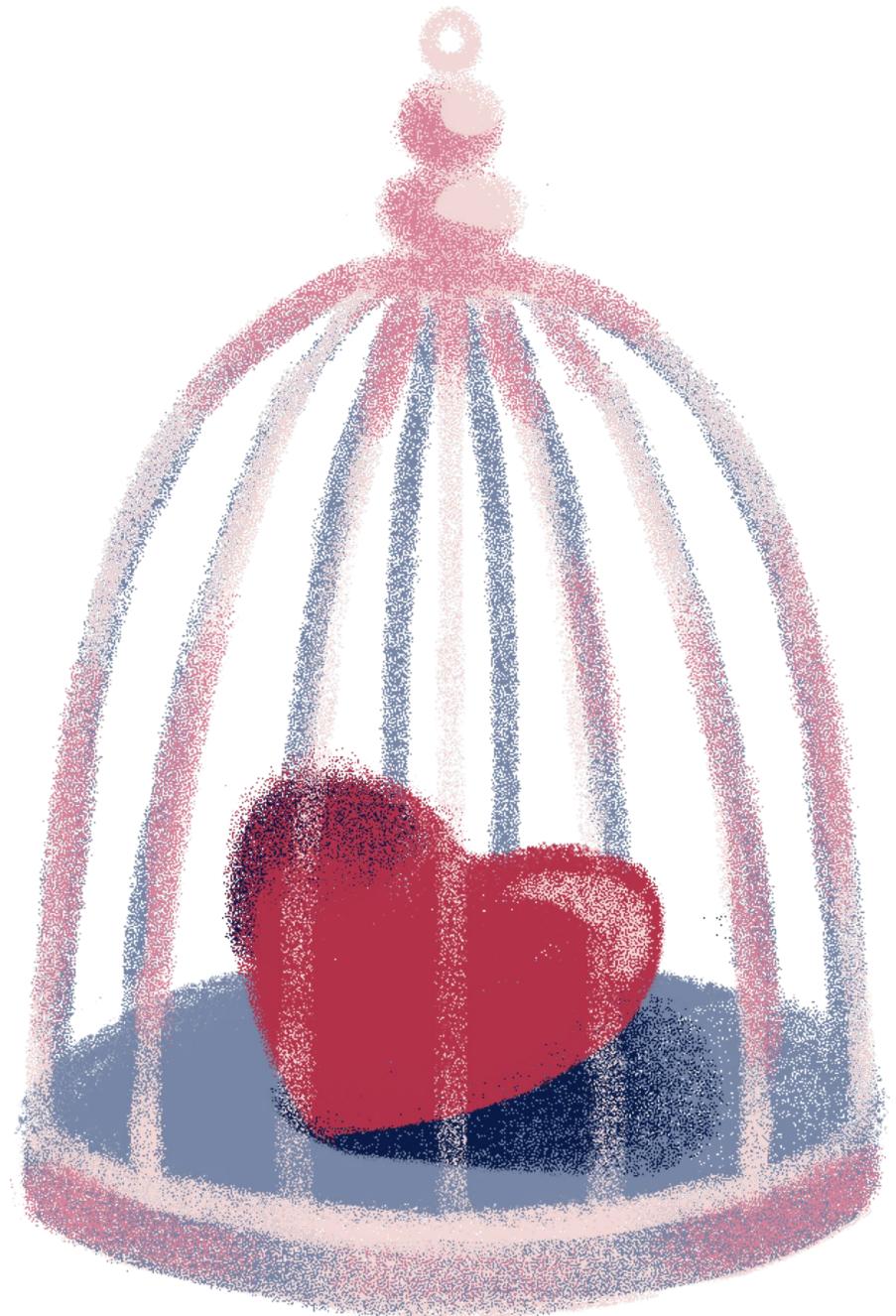


Table of Contents

01. 제작 동기

06. 주요기능 & User Flow

02. 비교 레퍼런스

07. 화면 설계

03. 팀원 업무 분담

08. 개발 중 발생한 이슈 및

04. 기술 스택

09. 행정별 느낀 점

05. ERD

10. 프로젝트 요약 및 향후

계획

02

하루의 감정이, 그냥 지나가지 않도록

02

지금 느끼는 감정은 분명하지만, 시간이 지나면 그 감정은
희미해집니다.

우리는 일기 속 감정을 조금이라도 더 오래 남기고,
다시 바라볼 수 있게 만들고 싶었습니다.

이 프로젝트는 AI를 통해 하루의 감정을 정리하고 쌓아가며,
작은 기록이 의미로 남길 바라는 마음에서 시작되었습니다.

01

제작 동기 및 특징

감정을 선택하는 기록이 아니라,
글 속에 담긴 감정을 이해하고 정리해주는 기록이 필요하다고 느꼈습니다.

① AI 감정 자동 분석

일기 내용을 기반으로
AI가 글 속 감정을 이해하고 분석합니다.
사용자는 감정을 직접 선택하지 않아도
자연스럽게 감정 기록이 완성됩니다.

② 감정 흐름 기록

분석된 감정은 사용자별로 저장되어
시간의 흐름에 따라 쌓입니다.
하루의 기록이 모여 나만의 감정
변화
과정을 한눈에 확인할 수 있습니다.

비교 레퍼런스



MOODA <무다>

여러 일기 기록 방식을 참고하여 사용자 경험을 분석해본 결과
일기를 기록할 때 우리는 주로 그날의 생각을 글로 남기거나
감정을 직접 선택(색상 선택)해야 했습니다. 기록은 쌓이지만,
시간이 지나도 내 감정이 어떻게 변해왔는지는 한눈에 보이지
않습니다.

팀원 업무 분담

이름	역할	책임 범위
김은혜	PM	프로젝트 기획 및 전체 관리, 테스트, 팀원 간 의견 조율, 유지보수, 인증시스템 구축
강혜정	SM (Scrum Master) 풀스택 개발자	프로젝트 아이디어 제안, 서비스 전체 구조 설계, 초기 공통 코드 베이스 구축 및 팀원 공유, CSS 보수, Git 서포트
김재혁	풀스택 개발자	Flask 라우팅, MariaDB 스키마 설계, 그래프 제작, Google Gemini API 연동 및 감정 분석 로직 개발, Cloudtype 배포, 산출물 관리
나상훈	풀스택 개발자	Git 버전 관리, 관련 자료 조사 및 정리, 아이디어 제안, MariaDB 연동, CSS 보수
임다빈	풀스택 개발자	HTML/CSS, 서버 작동 테스트, 아이디어 기획, git 버전 관리, MariaDB 연동

04

기술 스택

프론트엔드

 **Jinja2** : 템플릿 엔진

 **HTML/CSS** : 반응형 웹 디자인

 **Chart.js** : 데이터 시각화 (감정 추이 그래프 등)

백엔드

 **Flask Framework** : 웹 애플리케이션

 **Werkzeug** : 비밀번호 해싱, 쿠키 관리 등
프레임워크
유ти리티

데이터베이스

 **PyMySQL** : MariaDB 연결 드라이버

 **MariaDB** : 관계형 데이터베이스

버전 관리 & 배포

 **Git** : 소스 코드 버전 관리

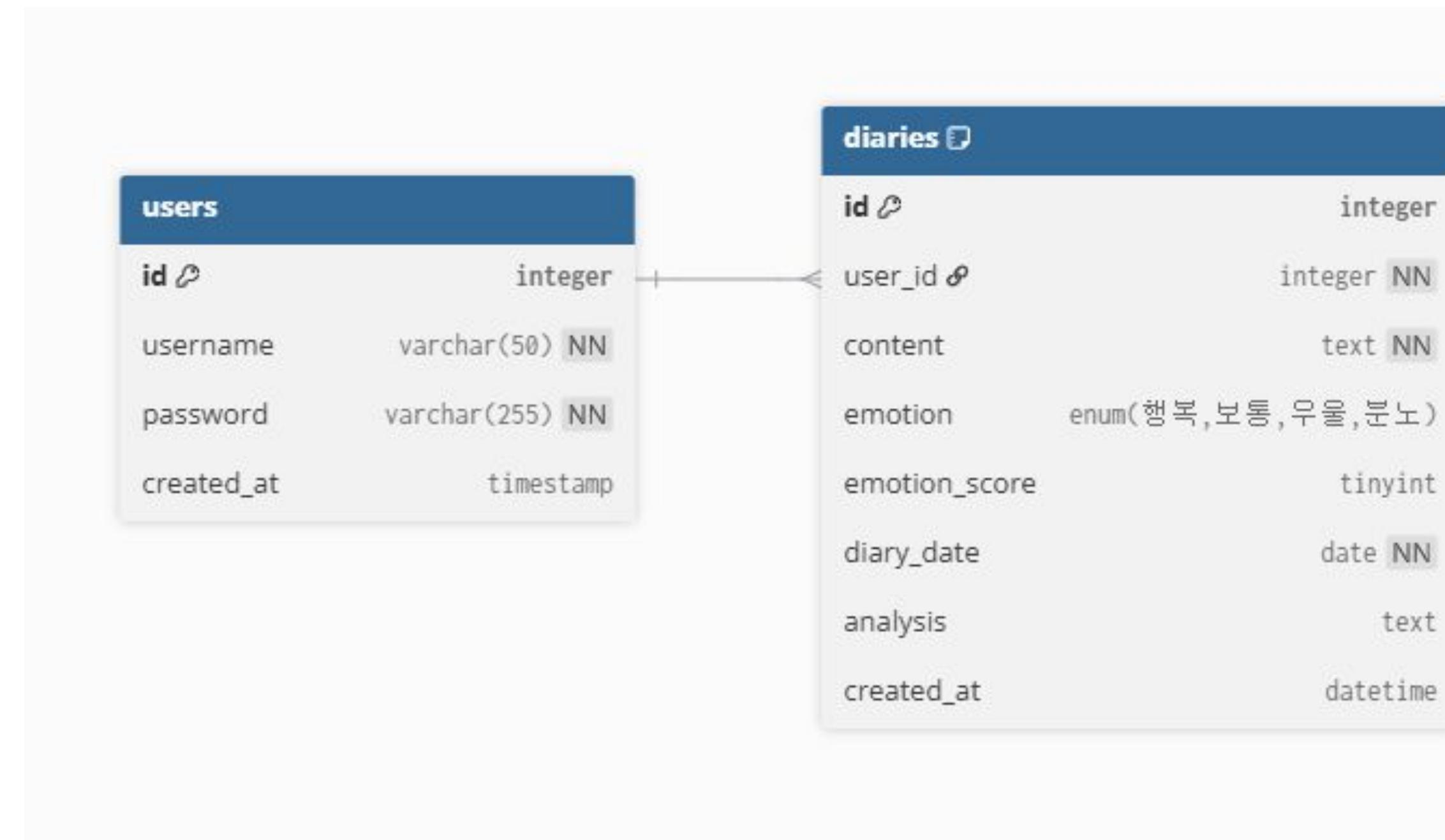
 **CloudType** : 클라우드 배포 플랫폼

AI / ML

 **Google Gemini API** : 감정 분석 엔진
(google-generativeai 라이브러리 사용)

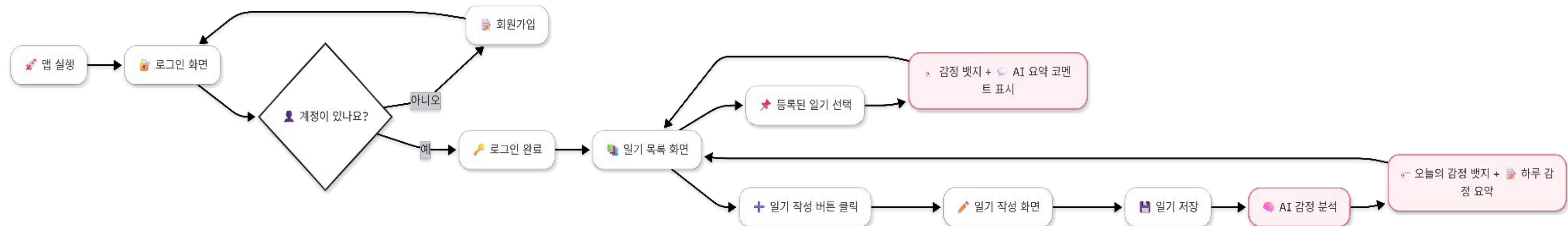
개발 도구

 **Visual Studio Code** : 통합 개발 환경 (IDE)



06

주요 기능 & User Flow



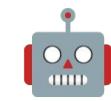
주요 기능



사용자 인증



일기 작성



AI 감정 분석



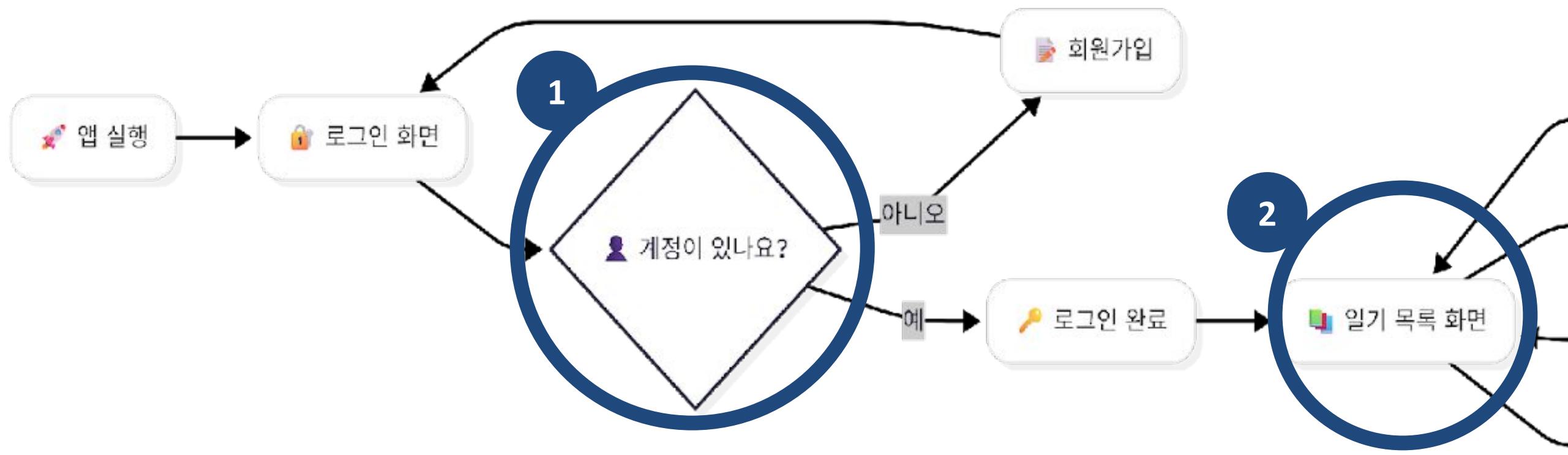
감정 코멘트 제공



기록 조회

07

주요 기능 & User Flow



01 사용자 인증

개인별 감정 기록을 관리하기 위한 기본
기능

회원가입과 로그인을 통해 사용자 구분
비밀번호는 해시 처리하여 안전하게 저장

구현

Flask 세션 기반 인증

사용자 정보는 MariaDB에 저장

02 기록 조회

쌓인 일기와 감정 기록을 확인

사용자별 일기 목록 조회

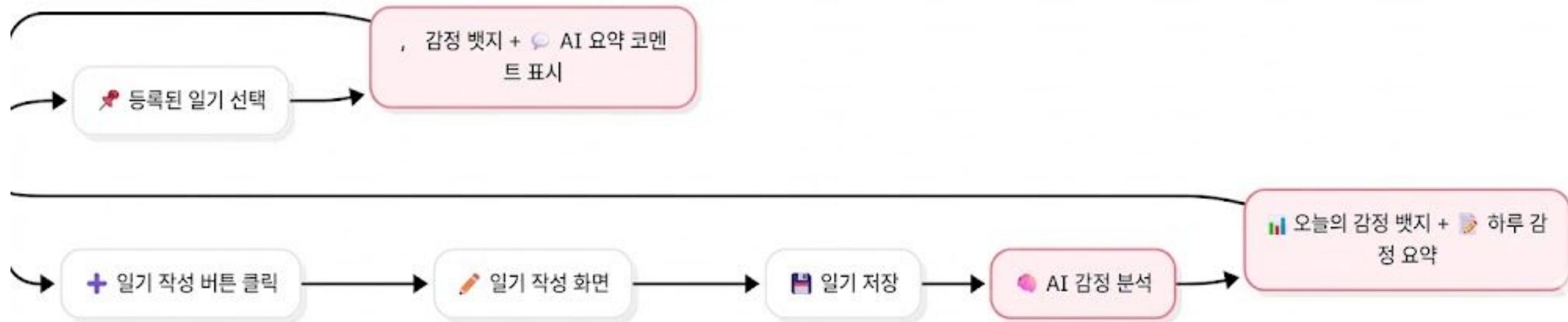
시간 흐름에 따른 감정 변화 확인

구현

사용자 ID 기준으로 일기·감정 데이터 조회

08

주요 기능 & User Flow



일기 작성

하루의 감정과 생각을 자유롭게 기록
날짜를 선택해 오늘의 감정을 글로
작성
형식 제한 없이 입력 가능

구현

일기 데이터를 DB에 날짜 기준으로
저장

AI 감정 분석

일기 내용을 바탕으로 현재 감정을
정리
작성된 일기 내용을 AI에 전달
감정을 4가지 감정 기준으로 분류

구현

Google Gemini API 호출
일기 텍스트를 분석해 감정 결과 반환

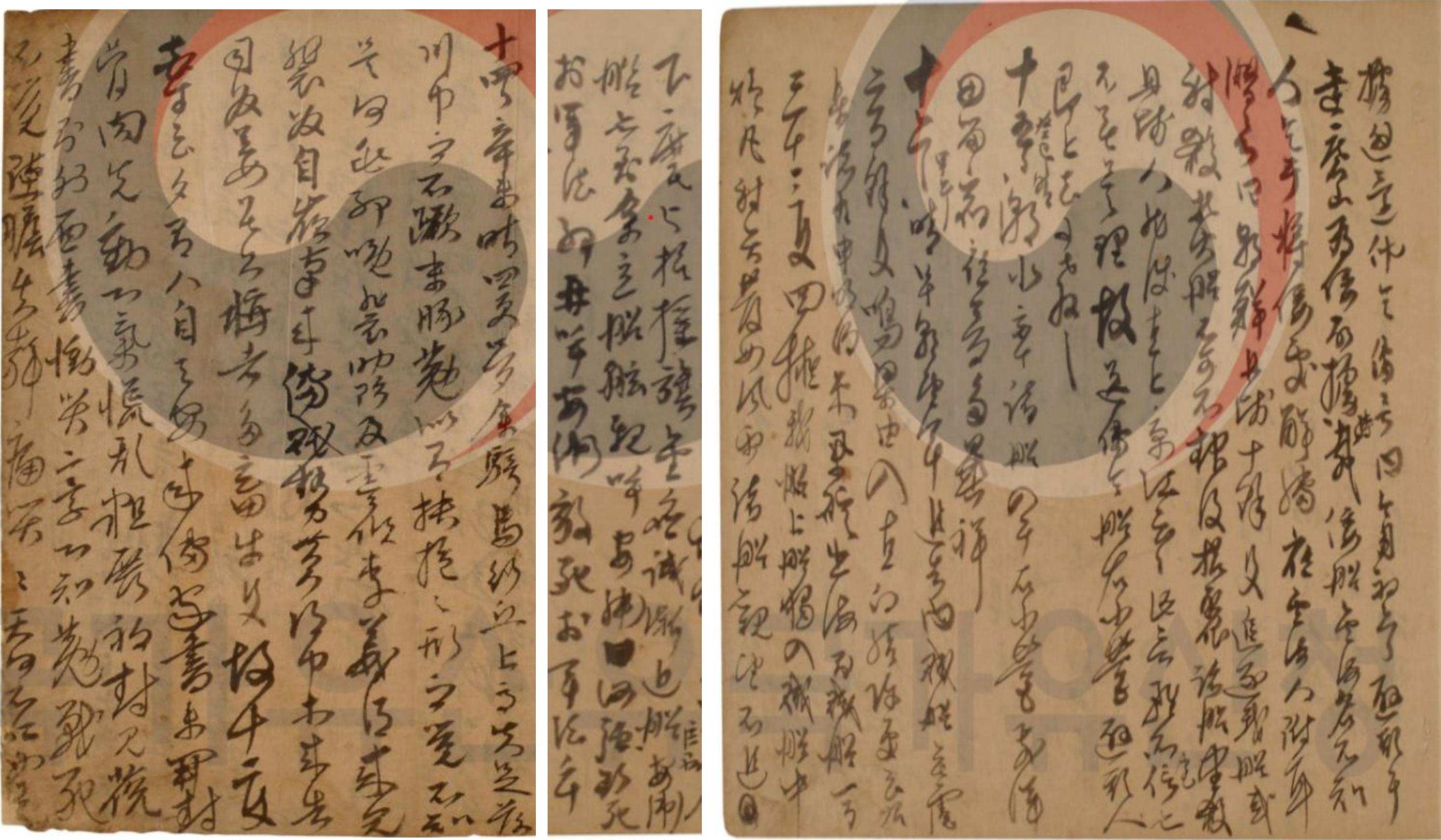
감정 코멘트 제공

분석된 감정에 대한 짧은 안내 제공
감정 결과와 함께
한두 문장의 짧은 AI 코멘트 제공

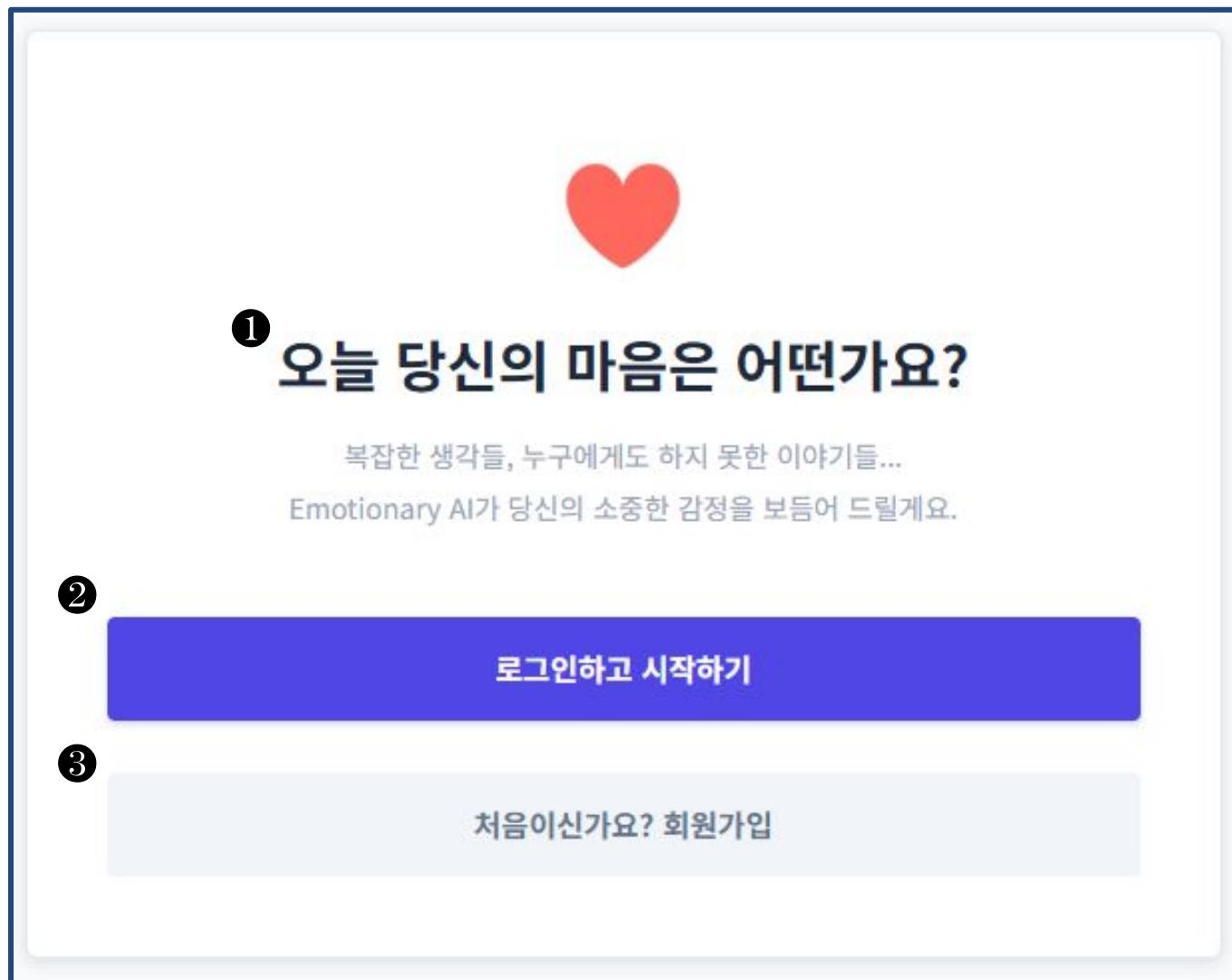
구현

AI 응답을 받아서 감정 요약 문장을
화면에 출력

시연 사례: 이순신,『난중일기』



화면 설계 (메인 페이지)



01. 프로그램의 간단한 소개

02. 로그인 페이지 이동 버튼

03. 회원가입 페이지 이동 버튼

10

화면 설계 (회원가입)

The screenshot shows a sign-up form with the following elements:

- 1.** 아이디 (ID): A text input field labeled "아이디를 입력하세요" (Please enter your ID).
- 2.** 비밀번호 (Password): A text input field labeled "비밀번호를 입력하세요" (Please enter your password).
- 3.** 가입하기 (Sign Up): A blue button labeled "가입하기" (Sign Up).
- 4.** 로그인으로 돌아가기 (Return to login): A link labeled "로그인으로 돌아가기" (Return to login) located at the bottom right.

At the top center of the form is a large red heart icon. Below it, the text "회원가입" (Sign Up) is displayed. A placeholder text "감정 기록을 시작해보세요" (Start recording your emotions) is visible in the background.

01. 아이디 작성 박스

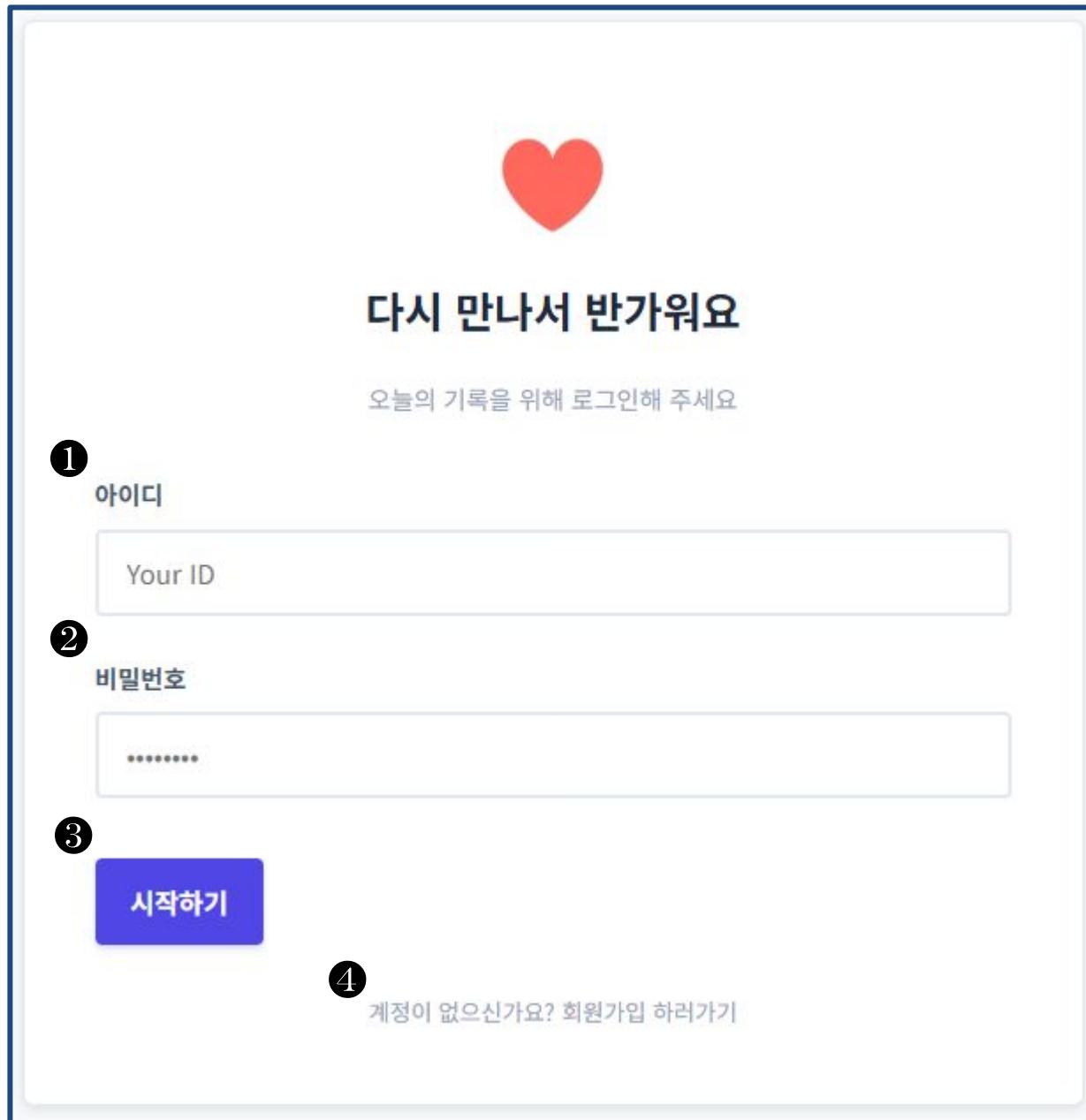
02. 패스워드 작성 박스

03. 회원가입 버튼

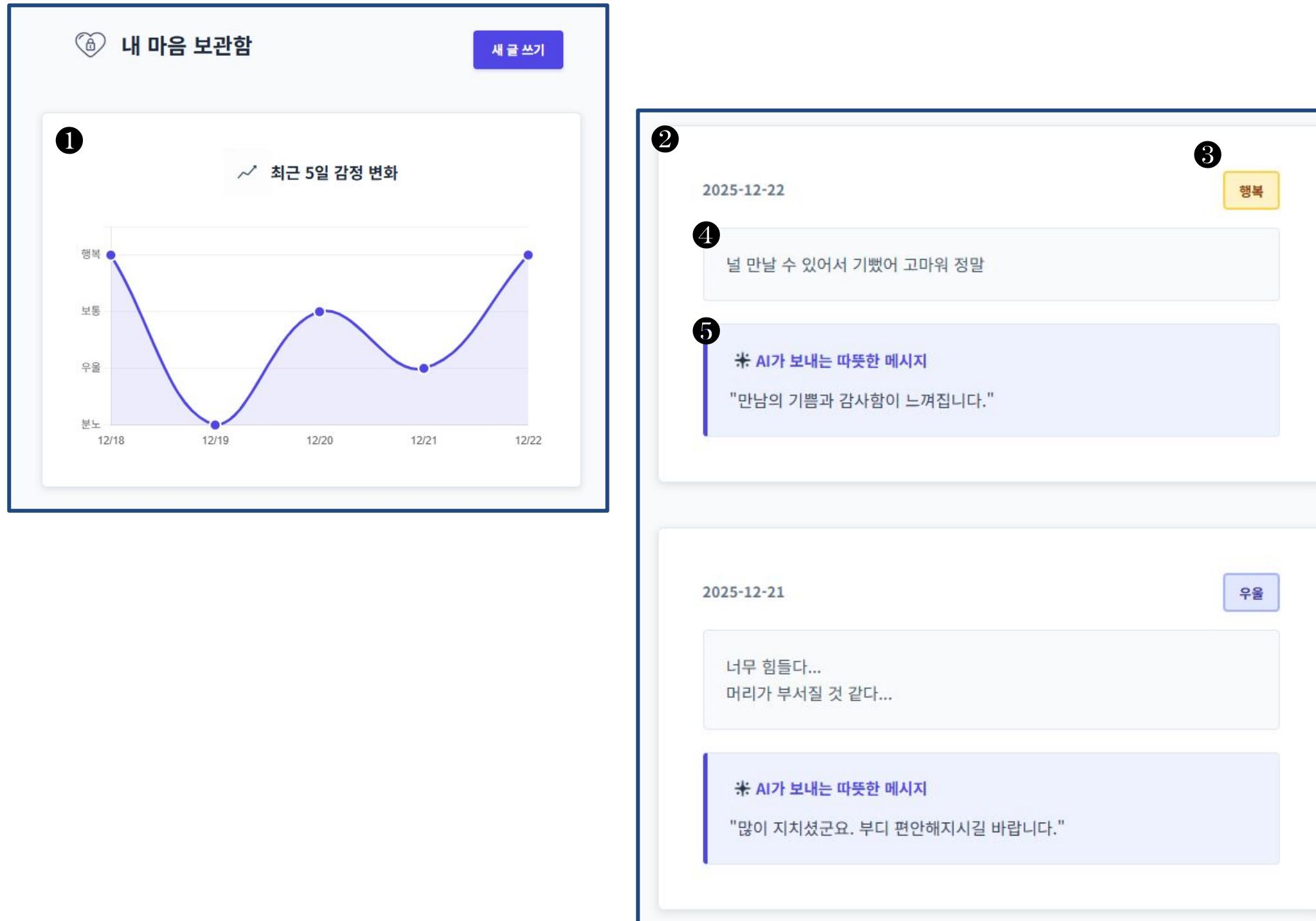
04. 로그인 페이지 이동 버튼

11

화면 설계 (로그인)

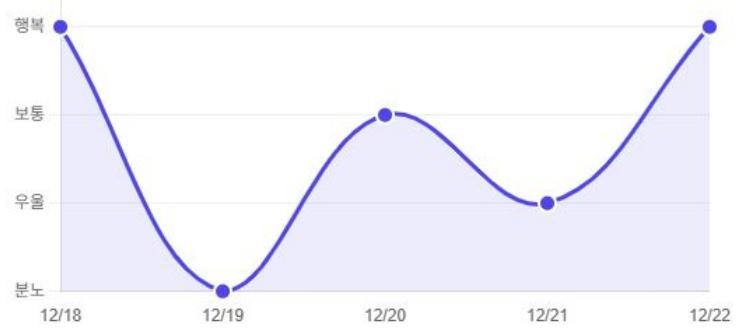
**01. 아이디 작성 박스****02. 패스워드 작성 박스****03. 로그인 버튼****04. 회원가입 페이지 이동 버튼**

화면 설계 (일기 리스트)



1

✓ 최근 5일 감정 변화



2

2025-12-22

3

행복

4
널 만날 수 있어서 기뻤어 고마워 정말

5
※ AI가 보내는 따뜻한 메시지
"만남의 기쁨과 감사함이 느껴집니다."

2025-12-21

우울

너무 힘들다...
머리가 부서질 것 같다...

※ AI가 보내는 따뜻한 메시지
"많이 지치셨군요. 부디 편안해지시길 바랍니다."

01. 최근 5일의 감정 분석 결과 그래프

02. 지난 일기 리스트

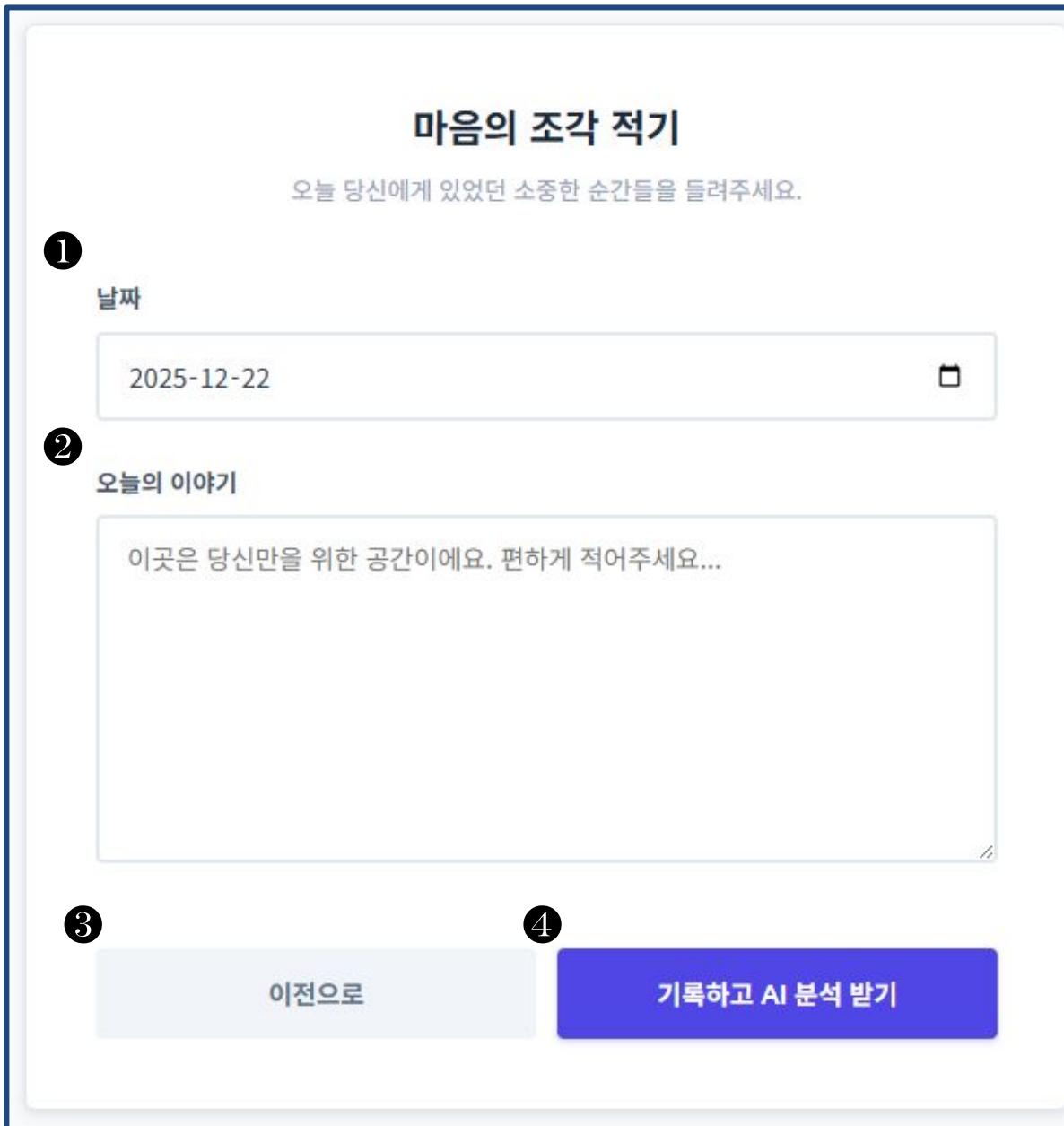
03. 일기 내용 기반 감정 분석 태그

04. 사용자가 작성한 일기 내용

05. 감정 분석 문구

13

화면 설계 (일기 작성)



01. 작성의 기준이 되는 날짜

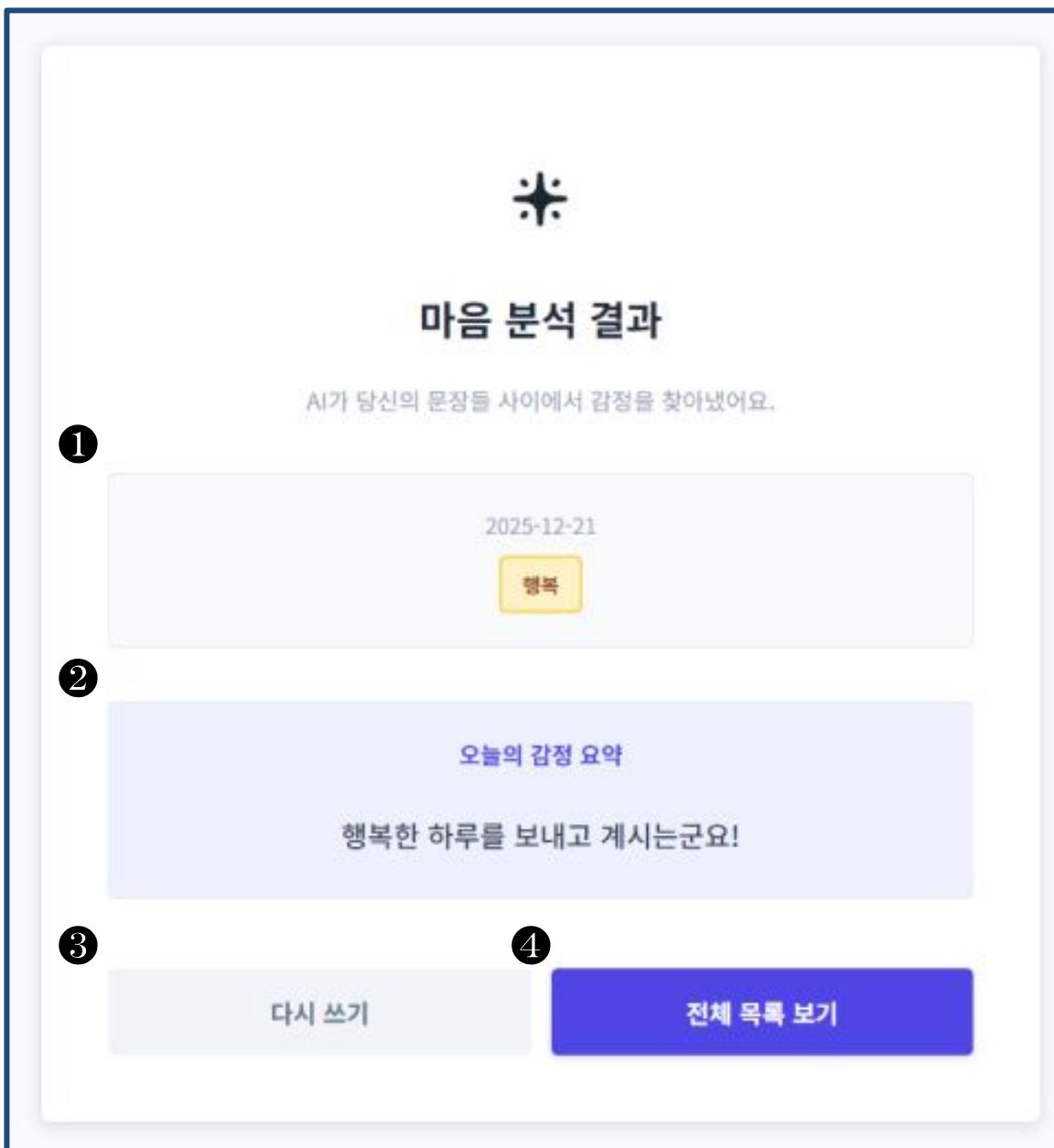
02. 일기 본문 작성란

03. 일기 리스트 페이지로 이동

04. 일기 작성 완료 및 AI 분석 받기 버튼

14

화면 설계 (일기 분석 결과)



01. 일기 내용 기반 감정 분석 태그

02. 감정 분석 문구

03. 일기 작성 페이지로 이동

04. 일기 목록 페이지로 이동

개발 중 발생한 이슈 및 해결

나상훈

개발 후 발표 당일 아침 일기장을 시연해야 했다.

그런데 공유받은 링크로 앱이 열리지 않았다.

앱의 생성자가 컴퓨터를 켜니 앱이 열렸다. 해결되었다.

강혜정

백엔드 보충이 필요한 구글 AI스튜디오 : 첫 빠대를 만들 때 구글 ai스튜디오를 활용했었다.

api보안문제로 파이썬파일이 아닌 타임스크립트 파일로 제공되어 이를 터널 프론트에 최적화된 프로그램이다보니 탄탄한 백엔드파일이 추가로 필요했었다. 이를 터널에서 받은 코딩을 프론트와 백엔드로 나누어 다시 재구성하여 빠대를 만들었다.

김재혁

api 연결 과정 : AI 기능을 개발 중 감정 결과가 계속 출력되지 않았기 때문에 다소 어려움이 있었다. 인터넷에서 여러 정보를 찾아가면서 모델과 api key도 바꿔보며 수정을 거듭한 결과 성공적으로 적용이 가능했다.

Cloudtype을 통한 배포 과정 : 아무리 배포를 하려고 해도 터미널에서 계속 오류가 발견되었다. Cloudtype은 Github의 파일을 기준으로 배포가 진행되기 때문에 환경 변수를 따로 입력해줘야하는데 그 과정에서 올바른 환경 변수를 찾는 데 시간을 많이 썼다. 관련 자료를 찾아보며 작업을 수행하여 배포를 적용시킬 수 있었다.

깃허브 연결 : 깃허브 사용이 익숙하지 않아보니, 풀/푸쉬/PR/이슈처리에 있어서 실행하는것이 쉽지 않았다.

조장님의 잘알려주셔서 잘 실행해보긴하였으나, 속적으로 알려주신부분을 복습하고, 계속 해보면서 손에 익혀야 할것같다.

저장소 구분역시 매우 중요하다는것을 배우고, 저장하기전에 어느 위치에 어떻게 이동하는지 체크를 하는것또한 중요하다.

김은혜

이번 프로젝트를 진행하며 Git 협업 규칙을 사전에 충분히 정하지 못한 점이 아쉬움으로 남습니다. 그로 인해 Main 브랜치에 직접 Push 되는 상황이 있었고 이를 통해 협업 과정에서 브랜치 권한과 작업 흐름을 명확히 합의하는 것이 중요하다는 점을 다시 한번 느끼게 되었습니다.

이 경험을 계기로, 다음 프로젝트에서는 Main 브랜치에 대한 직접 Push를 제한하고 모든 변경 사항을 PR 기반으로 검토한 뒤 병합하는 협업 방식을 적용하려고 공부중입니다

그리고 작업 의도를 좀 더 확실히 보여주기 위해 여러 검색을 한 결과 Hooky를 활용해 feat, fix 등 기능 키워드가 자동으로 반영되도록 하여 커밋 기록만으로도 작업 의도와 변경 내용을 명확히 파악할 수 있는 환경을 구축하고 싶습니다!

임다빈

인공지능 버전 : 빠르게 발전하며 변화하는 인공지능 덕에 예전 인공지능의 버전들을 새 인공지능으로 업그레이드 하는 과정이 필요했다.

타입 변환 : AI를 활용하여 코드를 만들어둬 앱으로 구현해 내는 과정에서 코드들을 하나하나 변환하여 수정하는 과정이 필요했다.

팀원 별 느낀 점

강혜정

얕은 지식이다보니 로직트리 설계부터 코딩까지 ai를 활용해도 쉽지 않았다. 하지만 배운것을 프로젝트를 통하여 활용해보자는 생각으로 실행해보니 각 ai툴에 대한 차이점도 알게되고, 로직트리와 MVC구조에 대해서도 이해할수 있었습니다.

김재혁

잘 모르던 툴을 사용해서 처음으로 프로젝트를 진행해봤다. 주제 선정 및 기획부터 개발과 배포까지 미니 캡스톤 과제지만 해볼 수 있어서 좋은 경험이었다고 생각한다. 또한 협업의 중요성을 다시 한 번 더 깨달을 수 있었던 것 같다.

임다빈

처음으로 본격적인 협업 프로젝트를 진행하면서 많이 배우고 성장할 수 있는 기회를 얻어서 기뻤습니다! 직접 기획부터 앱 구현까지 다양한 경험을 하며 한층 더 이해도가 늘어나는 좋은 경험이었습니다!!

김은혜

다양한 팀원들과 협업하며 여러 환경에서 발생하는 오류를 직접 경험할 수 있었고, 그 과정이 신기하고 재미있었지만 저 스스로 부족함을 많이 느꼈습니다. Git 협업 규칙 없이 작업하면서 충돌과 혼선이 발생했고, 이를 통해 사전 협업 규칙 설정의 중요성을 깨닫게 되었습니다.

현재는 하나의 app 파일에 DB 로직뿐만 아니라 요청 처리, 비즈니스 로직, 초기화 로직 등 여러 역할이 함께 포함된 구조이지만, 추후에는 App Factory 패턴을 활용해 필요할 때 생성할 수 있는 방식으로 구현해보고 싶습니다 Blueprint를 이용해 페이지와 기능을 역할별로 분리하는 구조로 리팩토링하고 싶습니다.

이러한 경험을 바탕으로 Flask의 구조적인 설계 방식에 대해 더 공부하고, 다음 프로젝트에서는 정돈되고 유지보수하기 쉬운 구조로 개발해보고 싶습니다.

나상훈

우리팀이 만든 일기앱을 직접 구동해보니, 성능이 뛰어나서 깜짝 놀랐다. 한국어, 한문, 영문을 모두 해독할 수 있었으며, 그 핵심 내용을 정확히 파악한 후 적당한 코멘트를 달아주었다. Gemini API를 활용한 앱의 성능이 이 정도일 줄은 몰랐다. 감탄했다.

프로젝트 요약 및 향후 계획

프로젝트 요약

AI 기반 감정 일기 분석 웹 애플리케이션 성공적 구현

- 회원가입과 로그인을 통한 인증 시스템 구현
- 작성된 일기를 기반으로 최근 5일간의 감정변화 그래프로 시각화

Gemini API를 활용한 정확하고 사용자 친화적인 감정 분석

기능 제공

- 행복, 보통, 우울, 분노의 4단계 감정 분석

팀 협력을 통한 성공적인 미니 캡스톤 프로젝트 완수

- 역할을 크게 나누지 않고 협력하는 과제 수행
- 지속적인 수정사항 및 고려사항을 보고하여 안정적인 프로젝트 수행

향후 계획

AI 분석 고도화

- 더 세밀한 감정 분류 (기쁨, 슬픔, 분노, 불안, 평온 등)
- 일기 분석 후 조언 및 문학 작품 추천

추가 기능

- 감정 캘린더 뷰
- 일기 검색/필터링, 내보내기 (PDF, 텍스트)
- 수정/삭제 기능 추가

성능 & UX 개선

- AI 분석 프롬프트 최적화
- 다크 모드, 접근성 개선
- 코드 구조 리팩토링

Q & A

Thank You for your attention