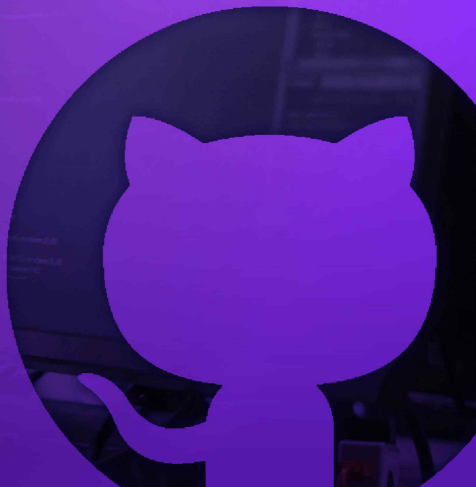


생성형 AI 기반 실용 웹서비스 개발자 양성 과정

코드베이스 기반 GitHub 이슈 분석 및 챗봇 가이드 플랫폼

프로젝트명 **AI[!]ssue** 팀명 | Mai-Nova

2025.06.17



개발 배경



GitHub
Copilot



CLAUDE
CODE

해외 빅테크 기업들의 움직임.. 심상치 않아!!

프로젝트 전체 코드베이스를 분석하여 복잡한 이슈를 해결해주는 서비스의 가능성

이것이 바로 AIssue의 시작점

<angle>



Baseline Widely available



[CSS](#) `<angle>` [자료형](#)은 각도의 값을 도, 그레이드, 라디안 또는 회전수로 표현합니다. [<gradient>](#) 나 일부 [transform](#) 함수에서 사용합니다..

| 오타 발견!!!

MDN Web Docs

Thanks for taking the time to contribute to this document. This document is a part of the MDN Web Docs project, which is a community effort to create a comprehensive, up-to-date, and accurate resource for web developers.

Getting started

Before contributing, please read the instructions for contributors. We also strongly encourage you to read the code of conduct. Community members are expected to follow the code of conduct at all times.

In addition to contributing to the content, you can also help by:

- Moderating discussions
- Fluency in English

Simple changes

If you want to make a small change to the content, you can do so by creating a pull request. The process is simple and straightforward. Here are the steps:

- Sign in to your GitHub account
- Navigate to the MDN Web Docs repository
- Find the file you want to edit
- Click the "Edit this file" button

From there, you can make your changes and submit a pull request. A pull request is a way to propose a change to the codebase. It allows you to discuss your changes with the maintainers and get feedback before they are merged into the main branch.

A pull request is created by clicking the "Create a pull request" button. You will be taken to the GitHub pull request page, where you can add a title, description, and labels to your pull request. Once you are ready, click the "Create pull request" button.

If you're not sure how to create a pull request, you can find more information in the [pull request guide](#).

Note: You may also contribute directly to the content by editing the file on GitHub. However, this is not recommended for large changes or changes that require discussion.

Forking and cloning

If you want to make a large change or work on a feature, you should fork the repository and clone it to your local machine. This allows you to work on your changes without affecting the main branch. Here are the steps:

- Fork the repository
- Clone the repository to your local machine
- Create a new branch for your work
- Make your changes
- Push your changes to the repository
- Create a pull request

Once you have created a pull request, you can track the progress of your work and see when your changes are merged into the main branch.

Mozilla

Translations

This document is available in multiple languages. If you are fluent in a language other than English, you can help by translating the document. Here are the steps:

- Spanish
- Japanese
- Korean
- Russian
- Simplified Chinese
- Traditional Chinese

Note: If you want to add a new language, you should first discuss it with the maintainers. You can do this by creating a pull request or by contacting the maintainers on the [MDN Web Docs mailing list](#).

Do not

In upstream code, you should not need to be concerned with the following:

- title "A"
- short title
- slug - no
- list

This guideline is for the content of the document, not the formatting. The guidelines are as follows:

Do not

At the time of the writing of this document, the guidelines are as follows:

- They provide
- They provide
- The upstream

Openness, diversity, and active participation are key to the success of the MDN Web Docs project. We encourage all community members to contribute and help make the project a better resource for web developers.

목차

1. MDN 번역 콘텐츠에 대한 일반 안내서
2. 첫 기여자들을 위한 안내서
3. yari 빠른 사용법
4. 여러 안내서
5. 자주 묻는 질문
6. 덧붙임

MDN 번역 콘텐츠에 대한 일반 안내서

이 문서는 한국(ko) 지역에 대한 번역 안내서입니다. 몇 가지 특정 안내서를 문서화하는 페이지를 추가하고 싶은 경우, 현재 경로에 추가하거나 [Locale Teams](#)에 문의해주세요.

마찬가지로 추가하려는 일반 안내서에 대한 아이디어가 있는 경우 환영합니다. Issue를 열고 문제에 대해 이야기해주세요. 다음은 한국어 번역 안내서 원칙의 주요 내용입니다.

어떻게 시작해야 할지 모르신가요? [MDN Discord #korean 채널](#) 혹은 [Google Groups \(yari-content-ko\)](#)으로 어려운 사항을 이야기해주세요!

첫 기여자들을 위한 안내서

MDN의 모든 로케일은 en-us 를 기준으로 번역이 진행되고 있습니다. 한국 로케일([translated-content](#))도 en-us (content)를 기반으로 번역이 진행되고 있습니다. 따라서, ko 와 en-us 문서 사이에 차이가 있다면, 미크업을 포함하여 en-us 문서를 기준으로 번역하여 translated-content 저장소에 기여할 수 있습니다.

[번역 안내서](#), [몰이 안내서](#) 및 [macro 안내서](#)를 숙지 후 기여해주세요.

관리자의 편집 허용

PR 생성 시 Create pull request 버튼 옆에 있는 Allow edits by maintainers 를 꼭 활성화해 주세요.

Prettier, Husky 설정

yarn install 명령어를 실행하여 git commit 실행 전에 Prettier와 Husky 활성화해 주세요.

PR 파일 개수

PR 생성시에 한국 리뷰어들은 라운드 로빈 방식으로 배정이 되고 있습니다. 한 리뷰어에게 너무 많은 리뷰가 몰리지 않도록 번역 PR의 경우 1개의 파일을 넘지 않는 것을 권장합니다.

사소한 변화 및 일괄적인 작업 PR에 대해서는 30개의 파일을 넘지 않는 것을 권장합니다.



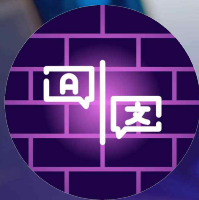
! 너무나도 많고 어려운 개발 문서..

개발 배경



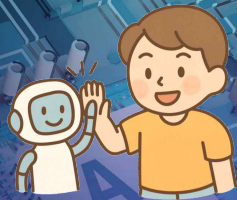
오픈소스의 복잡성

오픈소스 프로젝트의 문서는 복잡하고
관련 코드를 찾기 어렵습니다.



언어 장벽

영어나 다른 언어로 작성된 README/이슈/문서
등은
국내 개발자에게 진입장벽이 됩니다.



AI 기술 활용

AI와 RAG 기술을 활용하면, 이슈 분석 · 코드
추천 · 문서 요약 · 번역 등 반복적이고 어려운
작업을 자동화할 수 있습니다.

AI 기반 GitHub 이슈 분석 플랫폼

AI로 GitHub 이슈 분석을 더 스마트하게

AIssue는 AI를 활용하여 GitHub 저장소의 이슈를 분석하고, 관련 코드를 추천하며, 컨트리뷰션 가이드를 제공하여 개발자의 생산성을 향상시키는 서비스입니다.

무료로 시작하기 →

👁 자세히 알아보기

스크롤하여 더 보기

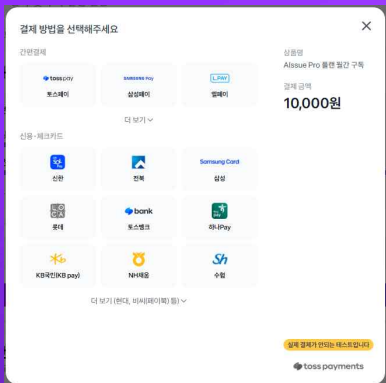


AI[💡]ssue란?

개발자들이 GitHub 저장소의 이슈를
쉽고 빠르게 이해하고, 관련 코드를 탐색

AI의 도움으로 해결 방안을 얻을 수 있도록
지원하는
AI 플랫폼

Alssue 데모 영상 (URL)



소에 대해 어떤 것이든 물어보세요. 컨트리뷰션 방법.

수많은 API가 계속해서 생기고 사라지
여전히 유효하고 잘 작동하는지, 그러
요?

public-apis는 수많은 API 목록을 커뮤니티
의견과 관련하여 저장소 파일 내용을 바탕으로

(Manual Curation): README.md 파일에 따르면
들에 의해 수동으로 큐레이션(manually curate
검토와 관리를 통해 유지됨을 의미합니다.

Contributor Responsibility): CONTRIBUTING.m
이드라인을 제시합니다.

- "추가하려는 API가 완전 무료 액세스 또는 최소한 무료 티
서비스 구매에 의존하지 않는지 확인하십시오."
 - "API에 적절한 문서(proper documentation)가 있는지 확인하십시오." 이러한 지침은 초기 기여 단계에
서 API의 품질과 접근성을 보장하는 데 중점을 둡니다.
- 3.
- 자동화된 링크 유효성 검사 (Automated Link Validation): CONTRIBUTING.md 파일에서 가장 구체적인 검
증 절차는 Pull Request(PR) 시 트리거되는 빌드 프로세스입니다.
- "Pull Request를 열면 프로젝트 내 모든 링크의 유효성을 확인하는 빌드가 트리거됩니다. 빌드가 완료되
면 빌드가 통과했는지 확인하십시오." 이는 PR이 병합되기 전에 목록에 추가되거나 수정된 API 링크가 유효
한지 자동으로 확인하여 깨진 링크(broken links)를 방지하는 중요한 역할을 합니다.

3. 코드 수정 제안 (Diff 형식)

아래 수정은 [현재 스크립트 파일의 실제 이름] 부분을 사용자의 실제 파일 이름(확장자 .py 제외)으로 대체해야 합니다. 예를 들어, 사용자의 파일이 rag_playground_app.py라면, [현재 스크립트
파일의 실제 이름]은 rag_playground_app이 됩니다.

파일 경로: [현재 스크립트 파일의 실제 경로 및 이름, 예: rag_playground_app.py]

HLS DIFF

```
--- a/[현재 스크립트 파일의 실제 이름].py
+++ b/[현재 스크립트 파일의 실제 이름].py
@@ -32,4 +32,4 @@
 app = Playground(agents=[rag_agent]).get_app()

 if __name__ == "__main__":
     serve_playground_app("rag_agent:app", reload=True)
+    serve_playground_app("[현재 스크립트 파일의 실제 이름]:app", reload=True)
```

4. 검토 및 주의사항

- 적용 점 확인사항:

- 가장 중요: 현재 이 코드가 저장된 Python 파일의 정확한 이름을 확인하십시오. 예를 들어, example_rag_agent_app.py와 같이 저장되어 있다면, 위 코드 수정 제안의 [현재 스크립트 파일의
실제 이름] 부분에 example_rag_agent_app를 정확히 입력해야 합니다.

- 예상되는 결과:

- 제안된 수정을 적용하고 스크립트를 다시 실행하면, Playground가 rag-agent의 세션을 올바르게 로드하고, UI를 통해 에이전트와 정상적으로 대화할 수 있게 될 것입니다. 404 Not Found 오
류는 더 이상 발생하지 않습니다.

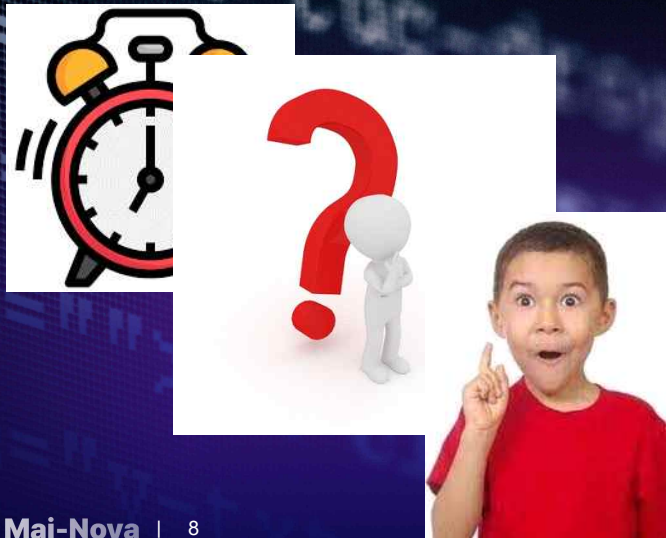
- 잠재적 부작용:

- 이 수정은 Unicorn의 모듈 로딩 메커니즘을 올바르게 맞추는 것이므로, 다른 기능적 부작용은 발생하지 않습니다.

- 추가적인 고려사항:

- "API에 적절한 문서(proper documentation)가 있는지 확인하십시오." 이러한 지침은 초기 기여 단계에
서 API의 품질과 접근성을 보장하는 데 중점을 둡니다.

핵심 가치 요약

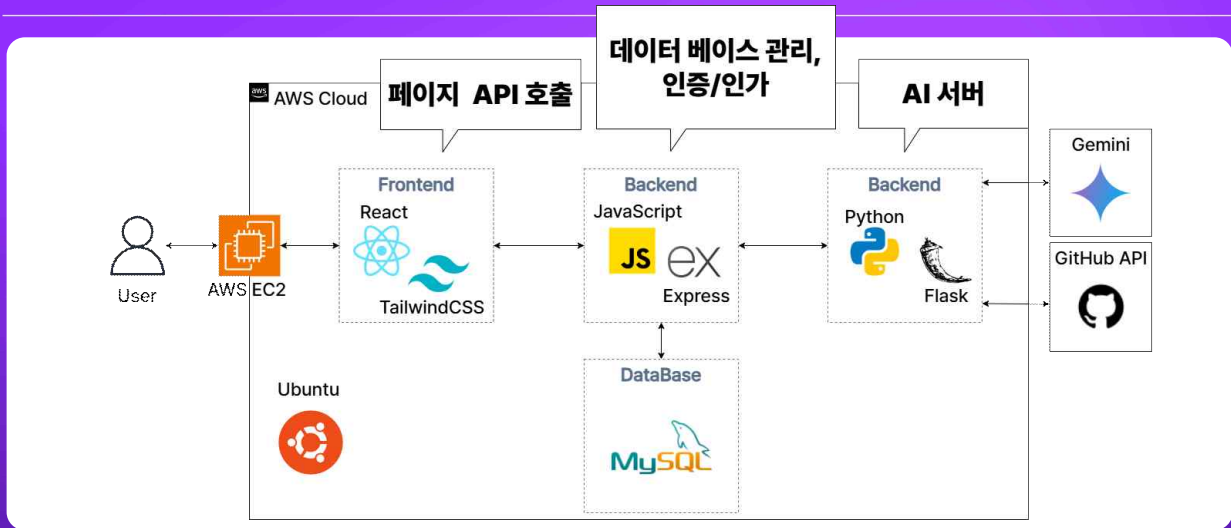


복잡한 이슈를 분석하는 시간을 크게 단축

코드를 찾아 헤매는 어려움을 해소

해결책을 제시해서 개발 효율 상승

시스템 아키텍처



저장소 등록 및 벡터화

대시보드

GitHub 저장소를 분석하고 이슈를 시로 해결해보세요

GitHub 저장소 분석

분석하고 싶은 GitHub 저장소의 URL을 입력하세요

Q. GitHub 저장소 URL 입력 (예: <https://github.com/username/repo>)

분석하기

분석 중인 저장소

typescript
microsoft/typescript
진행률: 65%

분석 중
완료 예정

분석이 완료되면 상태를 '완료'로 표시합니다.

react
facebook/react
진행률: 30%

분석 중
완료 예정

분석이 완료되면 상태를 '완료'로 표시합니다.

새로 분석된 저장소

my-project NEW
개인 프로젝트 저장소

비공개

🕒 분석 완료: 2023-05-18



코드를 고차원 벡터로 변환



유사도 기반 빠른 검색 지원



대규모 코드베이스 효율적 처리

저장소 등록 및 벡터화

사용자



저장소 분석 요청 (URL)

Express (BE)

ex

요청 수신, 유효성 검사

▼ GitHub 토큰 확인

GitHub API로 저장소 정보/파일 트리 조회

▼ 인덱싱 필요시

Flask에 인덱싱 작업 요청 (Repo URL, 콜백 URL)

▼ Flask로부터 콜백 대기

분석 완료/실패 알림

▼

DB 상태 업데이트 (완료/실패, 결과 저장)

Flask (AI/분석)



인덱싱 요청 수신

▼ 저장소 클론 및 파일 다운로드

(GitHub API 사용)

▼

코드 파싱 및 청킹

▼ 텍스트 임베딩

(Gemini API)

▼ 벡터 저장

FAISS 인덱스 생성/저장

▼ README 요약

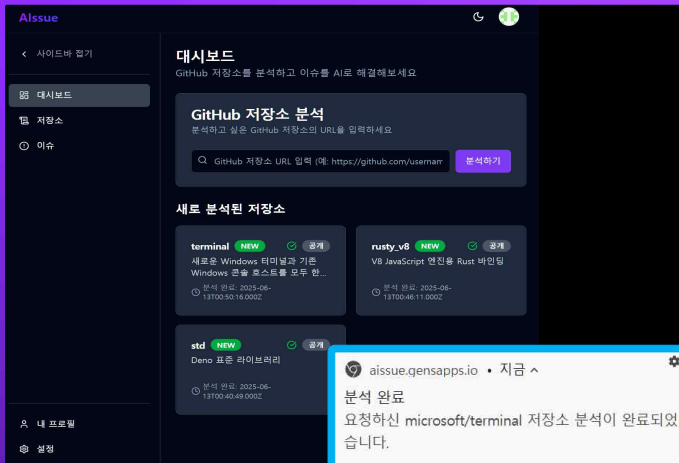
(Gemini API)

▼

작업 완료/실패 후 Express에 콜백

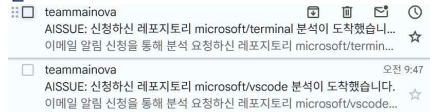
브라우저 및 이메일 알림

언제 끝나지?



- ✓ 저장소 분석은 시간이 걸리는 작업이기 때문에, 사용자가 마냥 기다리지 않도록 비동기 알림 시스템을 구축

분석이 완료되면 화면 우측 하단에 브라우저 알림이 뜨고, 등록된 이메일로도 분석 완료 알림을 보내주어 결과를 확인



AISSUE: 신청하신 레포지토리 microsoft/terminal 분석이 도착했습니다.
다. 받은편지함 x

✦ 이메일 요약



이슈 코드 매칭 및 AI 분석

핵심 요약:

코드 스니펫

스니펫 1

스니펫 2

```
import Issue from '../models/Issue.js';
import axios from 'axios';
import githubApiService from '../services/githubApiService.js'; // githubApiService 임포트

// Flask 서버 URL 설정
const FLASK_API_URL = process.env.FLASK_API_URL || 'http://localhost:3002';

// 여러 저장소의 이슈를 한 번에 조회
const getIssuesByRepoIds = async (req, res) => {
  try {
    const { repoIds, state, limit = 20, offset = 0, search } = req.query;

    if (!repoIds) {
```

controllers/issueController.js

57.9%

설명: 이 코드는 이슈와 57.9% 관련되어 있습니다.



요약

이슈 상세 페이지에 들어가면
AI가 이슈 내용을 자동으로



이 이슈와 관련된 코드 파일을
관련도 순으로 추천



관련도가 높은 코드 조각을
직접 보여줌

이슈 코드 매칭 및 AI 분석

3. 코드 수정 제안 (Diff 형식)

파일 경로: controllers/issueController.js

HLS DIFF

```
---
+++
@@ -1,2 +1,3 @@
 //
 con
 t
-   const { limit = 20, offset = 0 } = req.query;
+   const { limit = 20, offset = 0, search } = req.query; // search 파라미터 추가
-
+   const result = await Issue.selectRecentIssues(
+       userId,
+       parseInt(limit),
-       parseInt(offset)
+       parseInt(offset),
+       search // search 파라미터 전달
+   );

  if (!result.success) {
```



이슈 내용 요약

시는

업이나

이슈 코드 매칭 및 AI 분석

사용자



이슈 상세 페이지 분석

Express (BE)

ex

요청 수신 (repold, issueNumber)

▼ DB 조회

이슈 정보 및 저장소 정보(repoURL) 로드

▼ 분석 필요 시

Flask에 이슈 분석 요청 (이슈 데이터, repoURL)

▼ 분석 결과 대기

(Flask로부터 분석 결과 수신)

▼

DB에 결과 저장

▼

사용자에게 분석 결과 반환 (FE에 표시)

Flask (AI/분석)



이슈 분석 요청 수신

▼ AI 요약

Gemini API: 이슈 제목/본문 요약

▼ 관련 코드 검색

FAISS: 요약/본문 기반 유사 코드 검색

▼ 해결 제안

Gemini API: 이슈 정보 + 검색된 코드 기반 해결책 제안

▼

Express에 분석 결과 반환 (요약, 코드 스니펫, 해결 제안)

컨트리뷰션 가이드 AI 챗봇

안녕하세요! 저장소에 대해 어떤 것이든 물어보세요. 컨트리뷰션 방법, 코딩 컨벤션, PR 작성 방법 등에 대해 답변해 드릴 수 있습니다.

이 저장소의 컨트리뷰션 방법은?

AlIssue-BE-Express 저장소에 기여하는 방법은 CONTRIBUTING 파일과 README.md 파일에 상세히 설명되어 있습니다. 이 두 문서를 종합하여 기여 절차를 안내해드립니다.

1. 기여 전 확인 사항
- ...
2. 기여 워크플로우
- ...

이슈 해결하고 있는데 PR은 어떤 식으로 올려?

AlIssue-BE-Express 저장소에서 이슈를 해결하신 후 Pull Request(PR)를 올리는 과정은 CONTRIBUTING 파일에 상세히 안내되어 있습니다. README.md 파일의 기여 관련 지침과 함께 종합하여 PR 생성 절차를 설명해 드립니다. 이슈 해결 후 PR을 올리는 주요 절차는 다음과 같습니다.

1. 새 브랜치 생성 및 작업 (Make a Branch and Work)
- ...
2. 변경 사항 커밋(Commit Your Changes)
- ...

✓ 저장소별 AI 챗봇

- 저장소별로 AI 챗봇이 제공되며, 사용자는 자유롭게 질문을 남길 수 있습니다.

✓ 문서 기반 답변

- 챗봇은 README, LICENSE, CONTRIBUTING 등 저장소의 실제 문서를 바탕으로 답변을 생성합니다.

✓ 질문 예시

- 이 프로젝트의 코드 컨벤션은?
- PR 작성 방법 알려줘

컨트리뷰션 가이드 AI 챗봇

사용자



챗봇에 질문 입력

Express (BE)

ex

요청 수신 (repold, 메시지 목록)

▼ DB 조회

저장소 정보 로드 (fullName, 파일명 등)

▼ 컨텍스트 정보 포함

Flask에 질문 답변 요청 (repo_name, 메시지)

▼ AI 답변 대기

(Flask로부터 AI 응답 수신)

▼

DB에 대화 내용 저장 (질문, 답변)

▼

사용자에게 AI 답변 반환 (FE에 표시)

Flask (AI/분석)



질문 답변요청 수신

▼ 컨텍스트 로드

GitHub 문서 파일 내용 로드

▼ AI 답변 생성

FAISS: 요약/본문 기반 유사 코드 검색

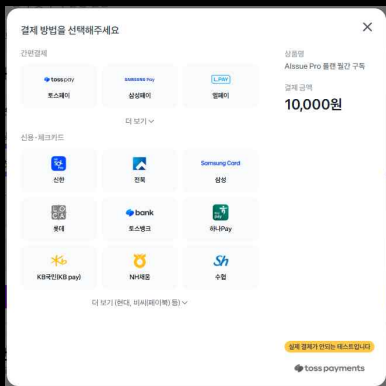
▼ 해결 제안

Gemini API:
(로드된 컨텍스트 + 사용자 질문) 기반 답변 생성

▼

Express에 AI 응답 반환

비즈니스 모델



- ✓ Aissue는 누구나 부담 없이 시작할 수 있는 무료 플랜과, 더 많은 분석과 고급 기능을 제공하는 전문가용 프로 플랜을 제공
- ✓ 사용자는 자신의 프로필 페이지에서 사용량을 직관적으로 확인할 수 있으며, 필요할 경우 토스페이먼츠를 통해 간편하게 프로 플랜으로 업그레이드 가능

팀원 소개



김지은@Pracrobo

브라우저, 이메일 분석 완료 알림
데이터베이스 ERD 설계
Express 저장소 추가



민유진@gnxm37

Github 로그인, JWT 인증/인가
토스페이먼츠 결제, Pro 구독 관리
Express 챗봇



최성호@c99-dev

CI/CD, 인프라 설계 / 구축
React 디자인, 레이아웃
Express README, 이슈
Flask Rag, 임베딩
Gemini, Github API 연동

AIssue AI 기반 GitHub 이슈 분석 및 코드 추천 플랫폼

아키텍처



프로젝트 개요

- 오픈소스 **이슈 분석**과 **코드 추천**을 AI로 자동화
- Google **Gemini LLM + RAG** (검색 증강 생성) 기반
- 이슈-코드 매칭, 저장소 요약, 챗봇, 한글 번역 등 지원

주요 기능

- **AI 이슈 분석** 이슈 요약, 관련 코드/파일 추천
- **저장소 요약** README/메타정보 요약, 한글 번역
- **AI 챗봇** 컨트리뷰션 가이드, 문서 기반 Q&A
- **대시보드** 내 저장소, 최근 본 이슈 관리

데이터 흐름

1. 사용자 **GitHub 로그인** 및 저장소 등록
2. 저장소 **코드 인덱싱** 및 AI 분석 준비
3. 이슈 상세 진입 시 **실시간 AI 분석 결과** 제공
4. 챗봇 Q&A, 최근 본 이슈, 대시보드 관리

AI 분석 방식

- 이슈 → **LLM 요약** → **코드 검색** → **해결 제안**
- RAG: 코드/문서 벡터화, 유사도 기반 추천
- 모든 답변/요약/번역은 **한국어**로 제공

기술 스택

- **프론트엔드**: React, Tailwind CSS
- **백엔드**: Express, MySQL
- **AI 서버**: Flask, Gemini API, FAISS

주요 장점

- AI 기반 **자동화**로 개발 생산성 향상
- 실제 코드/문서 기반 **정확한 추천** 제공
- **한국어 최적화** UI 및 AI 답변
- 확장성 높은 모듈형 **API 구조**

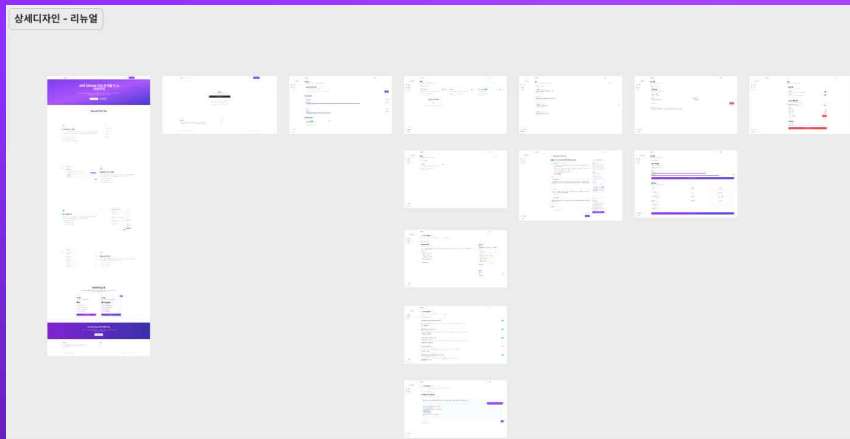
활용 예시

- 이슈와 관련된 코드/파일을 **AI 추천**으로 빠르게 탐색
- 영어 README/이슈를 **한글 요약**으로 쉽게 이해
- 챗봇에게 **프로젝트 컨벤션/가이드라인** 질문

비전

- 누구나 쉽게 오픈소스에 **기여**할 수 있는 환경 제공
- AI + RAG로 **글로벌 협업**의 언어 장벽 해소

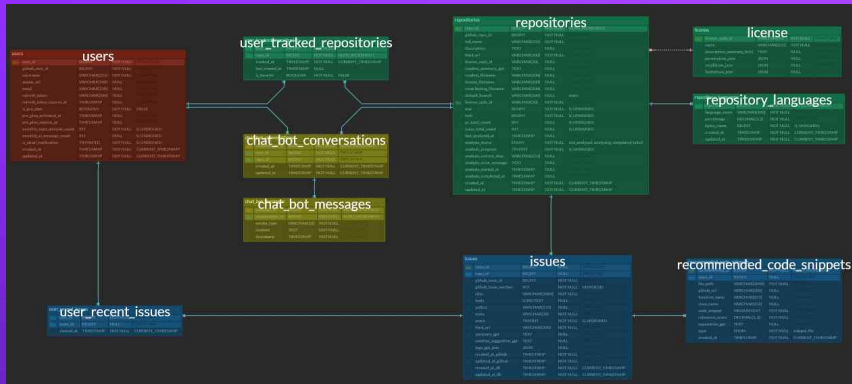
참고자료 – 화면 설계서 Figma



✓ 피그마를 통해 서비스의 전체적인 구조와 사용자 경험을 세심하게 설계

✓ 피그마 링크
<https://www.figma.com/design/fM5wa8nAQBB5PzPaD/mai-nova?node-id=0-1&t=DL2Mo7v8fYj5B4dU-1>

참고자료 – ERD ERDCloud



✓ ERDCloud를 통해 미리 스키마를 정의하고 저장 데이터를 설계

✓ ERDCloud 링크
<https://www.erdcloud.com/d/wYEhcNo8782h>

참고자료 – 프롬프트 aissue/readme.md

AI 프롬프팅 기법 및 구현

1. 번역 프롬프트 (Translation)

목적: 한국어 기술 질문을 영어로 정확히 번역하면서 코드 요소는 보존

사용된 프롬프팅 기법:

- Chain of Thought (CoT): 4단계 순차적 사고 과정 ([식별] → [유지] → [번역] → [형식])
- Few-shot Learning: 구체적 입력-출력 예시를 통한 학습 패턴 제공
- Role-playing: "Aissue의 전문 번역 AI 어시스턴트" 역할 부여로 전문성 강화
- Output Formatting: 최종 결과물만 출력하도록 하는 형식 제약 기법

실제 프롬프트 구조:

역할

당신은 Aissue의 전문 번역 AI 어시스턴트입니다. 프로그래밍 관련 한국어 질문을 정확하고 자연스러운 영어로 번역하는 것이 당신의 주요 임무입니다.

지시사항

1. **[식별]**: 한국어 텍스트에서 프로그래밍과 관련된 모든 고유 요소를 찾으세요. 여기에는 변수명, 함수명, 클래스명, 라이브러리명, 코드 블록 등 포함됩니다.
2. **[유지]**: 식별된 프로그래밍 고유 요소들은 번역하지 말고, 원문 그대로 정확히 유지하세요.
3. **[번역]**: 프로그래밍 요소를 제외한 나머지 자연어 한국어 문장을 명확하고 자연스러운 영어로 번역하세요.
4. **[형식]**: 최종적으로 번역된 영어 텍스트만 출력하세요. 다른 어떤 설명이나 사과, 원문 한국어 텍스트도 포함해서는 안 됩니다.

답변 구조

한국어 질문: '데이터프레임 `df`에서 `price` 컬럼의 값이 1000 이상인 행들만 선택해서 `high_price_df`라는 새 변수에 저장하는 파이썬 코드' 영어 질문: 'How do I write Python code to select rows from the DataFrame `df` where the `price` column is greater than or equal to 1000 and store them in a new variable named `high_price_df`?'



전략적인 프롬프팅으로 퀄리티 높은 응답을 이끌어내어 제품 품질 향상



Aissue Flask readme.md 링크

<https://github.com/Mai-Nova/Aissue-BE-Flask/blob/main/README.md>