

Phanes Editor

Phanes Editor 그리스 로마 신화 생산의 신

Team : 참자리들 팀

목 차

1. 프로젝트 개요	7 - 3
------------------	-------

1.1 소 개

1.2 배 경

1.3 시나리오

프로젝트 개요

1.1 소개

'Phanes Editor'는 별도의 프로그램 설치 없이 웹 브라우저만으로 실시간 코드 공유 및 동시 편집이 가능한 웹 기반 페어 프로그래밍 서비스입니다.

사용자들은 고유 링크를 통해 독립된 프로젝트에 참여하여, 마치 하나의 모니터와 키보드를 공유하는 것처럼 협업할 수 있습니다. 실시간 채팅과 코드 실행 기능까지 통합하여 원격 코드 리뷰, 기술 면접, 온라인 코딩 교육 등 다양한 상황에서 발생하는 협업의 장벽을 낮추는 것을 목표로 합니다.

1.2 배경

최근 비대면 업무 환경이 보편화되고 개발자 간의 원격 협업이 중요해지면서, 효율적인 페어 프로그래밍 도구에 대한 필요성이 증가하고 있습니다. 기존의 화면 공유 방식은 한쪽이 수동적인 관찰자 역할에 머무는 한계가 있으며, 로컬 개발 환경을 동기화하는 것은 번거롭고 시간이 많이 소요되는 작업입니다.

이러한 문제를 해결하기 위해 'Phanes Editor'는 웹 기반의 즉시성과 실시간성을 통해 물리적, 환경적 제약 없이 여러명의 개발자가 하나의 코드에 집중하고 함께 문제를 해결해나가는 몰입형 협업 경험을 제공하고자 합니다.

1. "화면 공유는 한쪽을 수동적인 관찰자로 만든다"는 한계

이것은 지금도 100% 현실입니다.

오늘날 대부분의 개발팀이 원격 회의 시 사용하는 줌(Zoom), 구글 미트(Google Meet), 슬랙(Slack) 등의 화면 공유 기능은 여전히 일방향적입니다.

- **비효율적인 커뮤니케이션:** 코드를 리뷰하거나 디버깅할 때, 코드를 보조하는 사람은 "거기 말고 조금만 더 위로 스크롤해주세요", "아니요, 53 번째 줄에 있는 변수명이요" 와 같은 구두 지시를 계속해야 합니다. 마치 옆에서 운전 훈수를 두는 것과 같아서, 직접 키보드를 잡고 고치는 것보다 훨씬 비효율적이고 답답합니다.

- **집중력 저하:** 관찰자 입장에 있는 사람은 직접 코드를 제어할 수 없기 때문에 시간이 지날수록 집중력이 떨어지고 수동적인 자세를 취하기 쉽습니다. 잠깐 다른 일을 하거나 딴생각을 해도 티가 잘 나지 않아, 진정한 의미의 '페어 프로그래밍'이 이루어지기 어렵습니다.

이 문제 때문에 **VS Code Live Share** 나 **JetBrains** 의 **Code With Me** 같은 IDE 확장 프로그램이 큰 인기를 끌었습니다. 이 툴들은 단순 화면 공유의 한계를 극복했지만, 아래에서 설명할 두 번째 문제를 완벽히 해결하지는 못합니다.

2. "로컬 개발 환경 동기화는 번거롭고 시간이 많이 걸린다"는 문제

이것 역시 **Docker** 와 같은 기술이 발전했음에도 불구하고 여전히 현실적인 장벽입니다.

간단한 알고리즘 문제 하나를 같이 풀거나, 특정 함수 하나를 리뷰하기 위해 두 개발자의 전체 프로젝트 환경을 완벽히 동기화하는 것은 여전히 부담스러운 일입니다.

- **OS 및 의존성 문제:** 한 명은 Windows, 다른 한 명은 macOS 를 사용할 경우 경로 문제나 자잘한 라이브러리 호환성 문제가 발생할 수 있습니다. 설치된 Node.js, Python, JDK 의 버전이 미세하게 달라 예상치 못한 오류가 생기기도 합니다.
- **설정의 복잡성:** 데이터베이스 연결 정보, 환경 변수(`.env`), IDE 의 고유 설정 등 코드 외적으로 맞춰야 할 것들이 너무 많습니다.
- **"배보다 배꼽이 더 큰 상황":** **Docker** 나 VS Code 의 **Dev Containers** 는 이 문제를 해결하는 훌륭한 방법이지만, 이는 개인의 개발 환경을 표준화하는 데 더 큰 목적이 있습니다. "지금 당장 이 코드 좀 5 분만 같이 봐줘" 라고 말하는 가벼운 협업 상황에서, 상대방에게 Docker 이미지를 빌드하고 컨테이너를 실행하라고 요구하는 것은 매우 번거로운 일이 될 수 있습니다.

결론: 'Phanes Editor'의 기회

정리하자면, 2025 년 현재에도 개발자들은 여전히 다음과 같은 선택지 사이에서 고민하고 있습니다.

1. **가장 빠른 방법:** 일단 줌으로 화면 공유부터 켜다. (하지만 답답하고 비효율적이다.)
2. **제대로 된 방법:** VS Code Live Share 나 Code With Me 를 쓴다. (하지만 둘 다 IDE 가 설치되어 있어야 하고, 한 명은 자기 PC 자원을 써서 호스팅해야 한다.)
3. **완벽한 방법:** 클라우드 IDE(GitHub Codespaces 등)를 쓴다. (하지만 세션 하나를 위해 컨테이너가 부팅되는 시간이 걸리고, 가볍게 쓰기엔 다소 무겁다.)

바로 이 지점에서 '**Phanes Editor**'와 같은 **경량 웹 기반 실시간 편집기**가 큰 가치를 가집니다. **설치도, 환경 설정도, IDE 종속성도 없이 그저 링크 하나만으로 즉시 페어 프로그래밍을 시작할 수 있다는 '가벼움'과 '즉시성'**이 다른 무거운 솔루션들과 차별화되는 강력한 무기가 될 수 있습니다.

- **비교할 만한 기존 서비스**

- **Replit**: 클라우드 기반의 뛰어난 IDE 이지만, 팀 협업 기능과 AI 기능은 유료(Pro 요금제 이상)로 제한되며, 언어 버전 변경이나 환경 커스터마이징이 어렵습니다. 또한, 무료 사용자는 리소스 제약으로 인해 실행 속도가 느리고, 초보자에게는 다소 복잡한 UI/UX 도 단점으로 작용합니다.
- **JetBrains Code With Me**: IntelliJ 등 강력한 로컬 IDE 환경을 그대로 공유할 수 있다는 장점이 있지만, 호스트와 참여자 모두 특정 IDE 를 설치해야 하며, 호스트의 PC 자원에 의존하기 때문에 호스트 컴퓨터가 꺼지면 협업이 중단됩니다. 무료 버전은 세션 시간과 인원 제한이 있어 실질적인 사용을 위해서는 유료 플랜이 필요합니다.
- '**Phanes Editor**'의 **차별점**: 위 서비스들과 달리 '**Phanes Editor**'는 설치가 필요 없는 순수 웹 기반 서비스로, 가볍고 빠르게 즉시 사용 가능한 페어 프로그래밍 본연의 기능에 집중하여 접근성과 편의성을 극대화합니다.

1.3 시나리오

- 실시간 코드 교육

- a. 서비스 웹 사이트에서 프로젝트를 생성
- b. 프로젝트가 생성되면, 고유한 초대 URL 이 자동으로 발급된다.
- c. URL 을 타고 B 가 프로젝트에 입장
- d. 도움이 필요한 A 가 실시간으로 자신의 코드를 작성
- e. B 가 문제점을 확인하고 직접 수정하면서 문제를 해결
- f. 작성이 완료된 코드를 실행하고 터미널에 실행된 결과를 같이 확인

- 작업 할당 및 관리

- a. 팀장의 팀원 초대
- b. 초대받은 팀원은 읽기만 가능하고 수정이 불가능

- c. 특정 파일/프로젝트에 대해 수정 권한 요청
- d. 팀원 별로 각자 다른 파일에 접근 권한 부여

- 프로젝트 생성 및 입장

- a. 사용자는 웹사이트에 로그인하여 프로젝트를 생성할 수 있다.
- b. 프로젝트 생성 시 시스템은 고유한 프로젝트 ID 와 초대 URL 을 생성한다.
- c. 초대 URL 을 가진 사용자는 로그인 없이도 프로젝트에 입장할 수 있다.
- d. 프로젝트에 입장한 사용자는 '게스트'로 자동 등록된다.
- e. 프로젝트 생성자는 기본적으로 '관리자 권한'을 가진다.

- 실시간 협업 및 코드 작성

- a. 프로젝트에 참여한 사용자는 코드 편집기에서 실시간으로 코드를 작성할 수 있다.
- b. 동일한 파일을 여러 사용자가 동시에 편집할 수 있다.
- c. 코드 편집 중인 위치는 사용자별로 커서 표시가 다르게 나타난다.
- d. 사용자가 타이핑할 때마다 변경 내용은 서버에 실시간으로 전송된다.
- e. 사용자의 변경 이력은 별도로 로그로 저장된다.
- f. 편집 중 파일은 일정 시간마다 자동 저장된다.

- 실행 및 결과 확인

- a. 사용자는 작성한 코드를 실행할 수 있다.
- b. 실행 결과는 공용 터미널에 창에 출력된다.
- c. 터미널은 프로젝트에 참여한 모든 사용자에게 동일하게 공유된다.
- d. 실행 기록은 로그로 저장되며, 프로젝트 종료 후에도 조회할 수 있다.

- 권한 및 접근 제어

- a. 프로젝트 생성자는 다른 사용자를 초대할 수 있다.

- b. 초대된 사용자는 기본적으로 '읽기 전용' 권한을 가진다.
 - c. 사용자는 특정 파일에 대해 '수정 권한 요청'을 할 수 있다.
 - d. 관리자(또는 소유자)는 각 파일/폴더 단위로 접근 권한을 설정할 수 있다.
 - e. 수정 권한이 없는 사용자는 편집기에서 입력이 제한된다.
 - f. 사용자는 '파일 단위' 또는 '프로젝트 단위'로 권한을 부여받을 수 있다.
- 팀 협업 및 작업 분배
 - a. 관리자는 사용자에게 특정 파일의 작업을 할당할 수 있다.
 - b. 할당된 작업은 사용자에게 별도 표시된다.
 - c. 사용자는 자신의 할당 작업 목록을 확인할 수 있다.
 - d. 사용자는 자신의 작업에 대한 진행 상태를 표시할 수 있다.
 - e. 관리자는 전체 작업 현황을 대시보드에서 확인할 수 있다.
 - f. 프로젝트는 팀 단위로 관리될 수 있으며, 팀 구성원은 프로젝트를 공유한다.
 - g. 하나의 사용자는 여러 개의 프로젝트에 참여할 수 있다.
 - h. 하나의 프로젝트에는 여러 명의 관리자가 존재할 수 있다.
 - i. 관리자는 다른 사용자의 권한을 변경하거나 제거할 수 있다.
 - j. 관리자는 프로젝트를 삭제하거나 공개 여부를 설정할 수 있다.
- 추가
 - a. 사용자는 프로젝트를 복제하여 새로운 프로젝트로 저장할 수 있다.
 - b. 사용자는 프로젝트에 '설명'을 추가할 수 있다.
 - c. 시스템은 비정상 종료 시, 자동으로 마지막 세션을 복원한다.
 - d. 사용자는 이전 세션의 편집 기록을 재생할 수 있다.
 - e. 사용자마다 개별 테마나 단축키 설정을 저장할 수 있다.