

# sangyi-tui

Zero Backend, All Local AI Agent Orchestration

<https://sangyi-tui.vercel.app>

## 프로젝트 개요

### 핵심 컨셉

- Zero Backend State:** 모든 데이터 로컬 저장
- 32 Agents Catalog:** 클라이언트 측 멀티 에이전트
- SQL-Queryable Local Data:** DuckDB WASM 활용
- Glassmorphism Design:** Apple 스타일 다크 모드

### 기술 스택

- Frontend:** Vanilla JS (프레임워크 없음)
- Database:** DuckDB WASM + IndexedDB
- Backend:** Vercel Edge Function (Stateless)
- API:** Anthropic SDK (SSE Streaming)



### Frontend

- Pure Vanilla JS
- SPA 아키텍처
- Zero dependencies
- 반응형 디자인

### Database

- DuckDB WASM (인브라우저)
- 10-table schema
- IndexedDB 영속성
- SQL 쿼리 지원

### Backend

- Vercel Edge (Stateless)
- SSE Streaming
- Session UUID 관리
- Token 비용 추적

# 핵심 기능

## 1. SSE 스트리밍 채팅

- 실시간 AI 응답 스트리밍 (Server-Sent Events)
- 마크다운 렌더링 + Tool Call 시각화
- 메시지별 톤튼 비용 추적 (input/output)

## 2. DuckDB WASM 데이터 레이어

- 10개 테이블 스키마: sessions, workspaces, messages, agents, skills, guides, teams, team\_members, tasks, cost\_tracking
- IndexedDB 영속성: 세션 간 안전한 데이터 보존
- SQL 쿼리 가능: 흐름 데이터 직접 분석

## 3. 워크스페이스 관리

- 멀티 워크스페이스 지원
- 인라인 이슈 변경
- 워크스페이스별 시스템 프로파일
- 삭제 확인ダイ얼로그

#### 4. 에이전트 카탈로그

- **32개** 에이전트 프리셋
- **16개** 스크립 라이브러리
- **7개** 가이드 문서
- 톤글 스위치로 활성화

#### 5. Agent Teams UI

- 팀 생성 및 멤버 추가
- Task CRUD 인터페이스
- 드래그 앤 드롭 지원
- 병렬 실행 계획

# 주요 기술 특징

기능	상세
@Mention 자동완성	퍼지 검색 + 키보드 네비게이션
비용 추적	작은 사용량 배지, 조합 가능한 시스템 프롬프트
모델 선택기	Sonnet 4.5 / Haiku 4.5 / Opus 4.6
세션 관리	crypto.randomUUID() 기반
Factory Reset	전체 데이터 초기화 기능
SSE 스트리밍	실시간 AI 응답 렌더링

## UI 디자인 철학



- Apple 스타일 **Glassmorphism**: backdrop-filter blur 효과
- 다크 모드: #0a0a0a 배경 + 그레이데이션 오브
- 레이아웃: 사이드바 + 메인 채팅 + 우측 패널 (설정)
- 애니메이션: 부드러운 전환 효과

# 파일 구조

```
sangyi-tui/
├── api/
│   └── chat.ts          # Vercel Edge Function (SSE Streaming)
├── public/
│   ├── js/
│   │   ├── app.js        # 메인 애플리케이션 컨트롤러
│   │   ├── db.js          # DuckDB WASM 데이터베이스 레이어
│   │   ├── chat.js        # 채팅 메시지 핸들링
│   │   ├── mentions.js    # @mention 자동완성
│   │   └── teams.js       # Agent Teams UI
│   └── css/
│       └── style.css      # Glassmorphism 스타일링
└── data/
    ├── agents.json       # 32개 에이전트 카탈로그
    ├── skills.json        # 16개 스킬 데이터
    └── guides.json        # 7개 가이드 문서
└── vercel.json          # Vercel 배포 설정
```

# sangyi-moru 와의 차별화

구분	sangyi-tui	sangyi-moru
핵심 스토리	Zero Backend 모든 데이터 훈련	Real Code Execution 클라우드에서 실제 실행
데이터 레이어	DuckDB WASM 브라우저 내부	PostgreSQL + Moru volumes
커버리 기능	32-agent 클라이언트 측 오퍼스트레이션	Live Web Preview 샌드박스에서 실행
고유 예모	SQL Console 훈련 데이터 커버리	Terminal + Preview 클라우드에서 실행
아키텍처	Edge Function (Stateless)	Full-stack (Stateful)
상태 관리	IndexedDB 100% 클라이언트	PostgreSQL 서버 측 영속성

핵심 차별점: sangyi-tui는 “안전한 로컬 AI 오케스트레이션”을 구현하며, sangyi-moru는 “클라우드 기반 실제 코드 실행”을 제공합니다. 두 프로젝트는 상호 보완적인 접근 방식입니다.

# 계획된 기능

## DuckDB SQL Console

- Cmd+K 단축키로 콘솔 오픈
- 흐름 데이터 직접 SQL 쿼리
- 결과 테이블 시각화
- 쿼리 히스토리 저장

예시:

```
SELECT agent_name, COUNT(*)  
FROM messages  
WHERE agent_name IS NOT NULL  
GROUP BY agent_name;
```

## Agent Team Execution

- 병렬 머티 에이전트 실행
- Task의 존성 그래프
- 실시간 진행 상황 모니터링
- 결과 집계 및 분석

시나리오:

- Agent 1: 데이터 수집
- Agent 2: 분석 (Agent 1의 존)
- Agent 3: 보고서 생성 (Agent 2의 존)

## 데모 URL

<https://sangyi-tui.vercel.app>

지금 바로 체험해보세요!

걍사장님나다

Zero Backend, All Local  
AI Agent Orchestration

[sangyi-tui.vercel.app](https://sangyi-tui.vercel.app)