

# 라이선스 관리 시스템

사용자 식별과 라이선스 발급 방안

# 문제배경

- AI 기반 STT 및 요약 프로그램을 개발하였음.
- 오직 결제를 완료한 사용자만 이용할 수 있도록  
솔루션 도입 필요.

# 사용자 식별 및 라이선스 발급

- 사용자 식별 방법

- 사용자는 이메일 주소를 이용하여 고유하게 식별된다.
  - 이메일이 중복되면 안됨(Unique key).

- 라이선스 코드 발급

- 사용자가 최초 결제하면, 자동으로 고유한 라이선스 코드가 발급된다.
  - 라이선스 코드는 프로그램을 활성화하는 데 사용되고 사용자 결제 내역과 연결

# 사용량 및 기간 관리

- 프로그램 사용과 관련된 정보를 관리해  
정해진 사용 범위 내에서 서비스 제공
  - **사용 시간**: 사용자가 프로그램을 실제로 사용한 시간
  - **사용 횟수**: 프로그램을 몇 번 사용했는지에 대한 기록
  - **남은 사용량**: 정해진 한도 내에서 남은 사용량
  - **라이선스 유효기간**: 라이선스가 유효한 기간 (예: 1개월, 12개월)
  - **다음 결제일**: 라이선스 갱신 또는 추가 결제가 필요한 날짜

# 스택기술

- Python : 주 개발 언어. 유지보수 쉬움. 느리지만 이해가 직관적.
- SQLite : 매우 가벼운 DBMS. 유지보수 쉬움. 기능은 최소화됨.
- C++ : 개발 언어. 속도가 중요한 모듈 있을시 사용하기.
- STT : Whisper V3 Turbo
- LLM : Gemini API, 추후 동 성능 로컬 LLM으로 가능하면 교체

# 동작 흐름

- 라이브클래스에서 상품 구입
  - 사용자가 결제를 완료하면, 이메일을 통해 고유 라이선스 코드 발급
- 프로그램 실행(Pro 버전)
  - 프로그램 실행시 라이선스 코드 입력해 프로그램 활성화
- STT, 요약 서비스 제공
  - 프로그램에서 라이선스 확인 후 STT, LLM 서비스 연결해 서비스 받음.
- 사용 내역 관리
  - 서버에서 사용 시간, 사용 횟수, 남은 사용량 등 사용 내역을 기록하고, 라이선스 유효기간 및 다음 결제일 정보 관리

# SQLite 스키마 설계 1

- 사용자 정보
  - **이메일**: 사용자 고유 식별자
  - 이름, 가입일 등 부가 정보(필요에 따라 추가)
- 라이선스 관련 정보
  - **라이선스 코드**: 최초 결제 시 발급되는 고유 코드
  - **유효기간**: 라이선스 시작일과 만료일
  - **다음 결제일**: 구독 갱신 또는 추가 결제 예정일
  - **사용 내역**:
    - 사용 시간, 사용 횟수, 남은 사용량

# SQLite 스키마 설계 2

- 사용 기록 (세션 로그)
  - 각 세션의 시작 및 종료 시간, 해당 세션의 사용 시간 등
- 결제(주문) 관련 정보
  - 주문 ID
  - **이메일**
  - 이름
  - 결제 금액
  - 결제 상태
  - 주문일



# SQLite 테이블 구조

- 테이블 4개로 구성
  - (1) Users 테이블
    - 사용자 정보
  - (2) Licenses 테이블
    - 각 사용자의 라이선스 정보
  - (3) Usage\_Logs 테이블
    - 프로그램 사용 세션 상세 기록
  - (4) Orders 테이블
    - 결제(주문) 내역 관리

# (1) Users 테이블

- CREATE TABLE IF NOT EXISTS users (
  - id INTEGER PRIMARY KEY AUTOINCREMENT,
  - email TEXT NOT NULL UNIQUE,
  - full\_name TEXT,
  - created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,
  - updated\_at DATETIME DEFAULT CURRENT\_TIMESTAMP
- );

## (2) Licenses 테이블

- CREATE TABLE IF NOT EXISTS licenses (
  - id INTEGER PRIMARY KEY AUTOINCREMENT,
  - user\_id INTEGER NOT NULL,
  - license\_code TEXT NOT NULL UNIQUE,
  - valid\_from DATETIME NOT NULL,
  - valid\_until DATETIME NOT NULL,
  - next\_payment\_date DATETIME,
  - usage\_time INTEGER DEFAULT 0, -- 누적 사용 시간 (초 단위)
  - usage\_count INTEGER DEFAULT 0, -- 프로그램 실행 횟수
  - remaining\_usage INTEGER, -- 남은 사용량 (설정된 경우)
  - created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,
  - updated\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,
  - FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE
- );

### (3) Usage\_Logs 테이블

- CREATE TABLE IF NOT EXISTS usage\_logs (
  - id INTEGER PRIMARY KEY AUTOINCREMENT,
  - license\_id INTEGER NOT NULL,
  - session\_start DATETIME NOT NULL,
  - session\_end DATETIME,
  - duration INTEGER, -- 세션 지속 시간 (초 단위, session\_end - session\_start)
  - created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,
  - FOREIGN KEY (license\_id) REFERENCES licenses(id) ON DELETE CASCADE
- );

## (4) Orders 테이블 (라이브클래스)

- CREATE TABLE IF NOT EXISTS orders (
  - id INTEGER PRIMARY KEY AUTOINCREMENT,
  - user\_id INTEGER NOT NULL,
  - license\_id INTEGER, -- 결제 시 해당 라이선스에 연결 (선택 사항)
  - order\_code TEXT NOT NULL UNIQUE,
  - amount REAL NOT NULL,
  - order\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,
  - payment\_status TEXT NOT NULL, -- 예: 'completed', 'pending', 'failed'
  - details TEXT, -- 추가적인 주문 관련 정보
  - created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,
  - FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE,
  - FOREIGN KEY (license\_id) REFERENCES licenses(id) ON DELETE SET NULL
- );