**관심사 기반 SNS 애플리케이션 개발 명세서 (Vercel & Supabase)**

제공해주신 상세 기획 문서를 바탕으로, Vercel과 Supabase를 활용한 \*\*'관심사 기반 SNS 애플리케이션'\*\*의 바이브 코딩(Vibe Coding)을 위한 개발 단계별 명세서를 작성했습니다.

**🎯 1. 프로젝트 개요**

* **프로젝트명**: 관심사 기반 SNS 애플리케이션
* **목표**: 사용자가 관심사 기반 커뮤니티에 참여하고 AI 큐레이터가 제공하는 정보를 통해 소통하는 웹 서비스 구축
* **핵심 기술 스택**:

**Milestone 1 (사용자 온보딩 및 인증) 구현에 사용된 기술:**

**\* 프론트엔드 프레임워크: React (Next.js 프레임워크 내에서 사용)**

**\* 언어: TypeScript**

**\* 스타일링: Tailwind CSS**

**\* 백엔드 서비스 (인증 및 데이터베이스): Supabase (BaaS - Backend as a Service)**

**\* 배포 플랫폼 (예정): Vercel (Next.js 애플리케이션 배포에 최적화)**

**전체 프로젝트에 제안된 기술 스택:**

**\* 프론트엔드: React (TypeScript) with Next.js, UI 라이브러리 (예: Bootstrap, Material Design)**

**\* 백엔드: Node.js (Express.js, TypeScript) 또는 Python (FastAPI) (Supabase를 백엔드로 활용할 경우, 커스텀 백엔드 필요성은 줄어들 수 있습니다.)**

**\* 데이터베이스: Supabase (PostgreSQL 기반)**

**\* 실시간 통신: WebSocket (Supabase Realtime 기능 활용 가능)**

**\* AI/NLP: LLM API 연동 (예: OpenAI API)**

**\* 배포: Vercel**

### 3. 개발 마일스톤 (Milestones)

개발 과정을 총 6개의 마일스톤으로 나누어 진행합니다.

#### Milestone 1: 사용자 온보딩 및 인증 구현

* **목표**: 사용자가 서비스에 가입하고, 관심사를 선택하여 프로필을 생성하는 전체 흐름을 완성합니다.
* **주요 기능 및 개발 포인트 (PDF, p. 3 참조)**:
  1. **시작 화면 (/)**: '시작하기' 버튼을 포함한 랜딩 페이지.
  2. **이메일 입력 및 검증 화면 (/login)**:
     + **Frontend**: 이메일 형식 유효성 검사.
     + **Backend (Supabase)**: auth.users 테이블을 조회하여 이메일 중복 여부를 확인하는 API(Edge Function) 구현.
  3. **관심사 선택 화면 (/onboarding/interests)**:
     + **Frontend**: interests 테이블에서 가져온 관심사 목록을 버블/태그 형태로 표시. 사용자가 선택한 관심사를 상태(State)로 관리.
     + **Backend**: 미리 정의된 관심사 목록을 제공하는 API.
  4. **닉네임 설정 화면 (/onboarding/profile)**:
     + **Frontend**: 닉네임 유효성(글자 수 등) 검사.
     + **Backend**: 이메일, 선택된 관심사, 닉네임을 받아 사용자 계정을 최종 생성하는 가입 API(Edge Function) 구현.
       - auth.users에 사용자 생성.
       - profiles 테이블에 닉네임과 함께 프로필 정보 저장.
       - profile\_interests 테이블에 선택한 관심사 저장.
       - 성공 시 JWT 토큰을 클라이언트에 전달.

#### Milestone 2: 채팅방 및 실시간 대화 기능 구현

* **목표**: 사용자가 관심사 기반 채팅방에 자동으로 참여하고, 실시간으로 메시지를 주고받는 핵심 기능을 구현합니다.
* **주요 기능 및 개발 포인트 (PDF, p. 2, 4 참조)**:
  1. **채팅방 자동 배정 로직**:
     + **Backend**: 로그인 성공 후, 사용자의 profile\_interests를 기반으로 chatrooms에 자동으로 참여(participants 테이블에 추가)시키는 로직 구현.
  2. **채팅방 목록 뷰 (/chat)**:
     + **Frontend**: 사용자가 참여 중인 채팅방 목록을 표시. 마지막 메시지, 안 읽은 메시지 수를 표시.
     + **Backend**: participants 테이블을 조회하여 사용자가 속한 채팅방 목록을 가져오는 API 구현.
     + **Realtime**: Supabase Realtime을 구독하여 새 메시지 도착 시 목록이 실시간으로 업데이트되도록 구현.
  3. **개별 채팅방 뷰 (/chat/[id])**:
     + **Frontend**: 메시지 스크롤 영역, 메시지 입력 필드 UI 구현.
     + **Backend**:
       - 화면 진입 시 이전 대화 기록(messages 테이블)을 불러오는 API.
       - 메시지 전송 요청을 처리하는 API.
     + **Realtime**: WebSocket(Supabase Realtime)을 통해 특정 채팅방의 메시지를 실시간으로 수신하여 화면에 렌더링.

#### Milestone 3: AI 큐레이터 백그라운드 기능 개발

* **목표**: 비동기적으로 작동하는 AI 큐레이터의 핵심 기능(콘텐츠 제공, 요약, 발행)을 서버 단에서 구현합니다.
* **주요 기능 및 개발 포인트 (PDF, p. 5 참조)**:
  1. **Supabase Edge Functions 개발**:
     + **외부 기사 요약 및 공유**: 채팅방의 최근 대화 내용을 LLM API에 전달하여 관련 아티클을 추천받고, messages 테이블에 'ai-curator' 타입으로 메시지를 전송하는 함수.
     + **대화 내용 요약**: 특정 채팅방의 대화 로그를 LLM API로 보내 핵심 내용을 요약하는 함수.
     + **토론 활성화 질문**: 대화가 없을 때, 주제와 관련된 질문을 생성하여 채팅방에 전송하는 함수.
  2. **스케줄링 설정 (Supabase pg\_cron)**:
     + 매일 자정, 대화 요약 함수를 실행하여 결과를 summaries 테이블에 저장하도록 스케줄링.
     + 1시간 이상 대화가 없는 채팅방을 감지하여 토론 활성화 함수를 실행하도록 스케줄링.
  3. **화면 표시**:
     + **Frontend**: 채팅방에서 message\_type이 'ai-curator'인 경우, 일반 메시지와 다른 UI로 렌더링.

#### Milestone 4: 피드 기능 구현 (인기글, 구독글)

* **목표**: 사용자들이 다양한 콘텐츠를 탐색할 수 있는 인기글 피드와 개인화된 구독글 피드를 개발합니다.
* **주요 기능 및 개발 포인트 (PDF, p. 4 참조)**:
  1. **인기글 피드 뷰 (/home/popular)**:
     + **Frontend**: 여러 채팅방의 인기글을 뉴스피드 형태로 보여주는 UI. 좋아요, 댓글, 저장, 공유 등 액션 버튼 포함.
     + **Backend**: 좋아요 수, 댓글 수 등을 기준으로 messages 테이블에서 인기글을 집계하여 제공하는 API 구현.
  2. **구독글 피드 뷰 (/home/subscribed)**:
     + **Frontend**: 인기글 피드와 유사한 UI로, AI 큐레이터가 생성한 요약글을 보여줌.
     + **Backend**: 사용자가 참여 중인(participants 테이블) 채팅방의 summaries 테이블 데이터를 조회하여 제공하는 API 구현.

#### Milestone 5: 프로필 관리 및 활동 점수 시스템

* **목표**: 사용자가 자신의 프로필을 수정하고, 서비스 내 활동 내역을 확인할 수 있는 기능을 구현합니다.
* **주요 기능 및 개발 포인트 (PDF, p. 6 참조)**:
  1. **프로필 관리 기능 (/profile/edit)**:
     + **Frontend**: 닉네임, 관심사 태그를 수정할 수 있는 UI 구현.
     + **Backend**: 변경된 닉네임과 관심사 정보를 업데이트하는 API 구현. 닉네임 중복 검사 로직 포함.
  2. **나의 활동 점수 기능 (/profile/activity)**:
     + **Frontend**: 총 활동 시간, 대화 횟수 등 지표를 시각화하여 보여주는 UI.
     + **Backend**:
       - 사용자 활동(메시지 전송 등)이 발생할 때마다 activity\_logs 테이블에 기록.
       - 매일 자정, 스케줄러(pg\_cron)를 통해 activity\_logs를 집계하여 사용자별 활동 점수를 계산하고 별도 테이블에 저장하는 배치 작업 구현.
       - 활동 점수를 조회하는 API 구현.

#### Milestone 6: 관리자 페이지 개발

* **목표**: 서비스 운영 및 모니터링을 위한 관리자용 웹 페이지를 구축합니다.
* **주요 기능 및 개발 포인트 (PDF, p. 7-8 참조)**:
  1. **대시보드**:
     + 총 가입자 수, DAU 등 핵심 지표를 시각화.
     + 필요한 통계 데이터를 제공하는 데이터 집계 API 구현.
  2. **사용자 관리**:
     + 사용자 목록 조회, 검색, 필터 기능.
     + 사용자 계정 정지, 정보 수정 등 CRUD API 구현.
  3. **콘텐츠 관리**:
     + 신고된 메시지 목록(reports 테이블)을 조회하고 처리하는 기능.
     + 부적절한 메시지를 삭제(Soft Delete)하는 기능.
  4. **활동/포인트 관리**:
     + 활동 점수 계산에 사용되는 가중치를 설정하는 UI 및 로직.
     + 특정 사용자에게 수동으로 포인트를 지급/회수하는 기능 및 로그 기록

, Vercel과 Supabase를 활용하여사용자가 관심사 기반 커뮤니티에 참여하고 AI 큐레이터가 제공하는 정보를 통해 소통하는 웹 서비스 구축하는데 개발 과정을 Milestone 1: 사용자 온보딩 및 인증 구현(완료), Milestone 2: 채팅방 및 실시간 대화 기능 구현, Milestone 3: AI 큐레이터 백그라운드 기능 개발, Milestone 4: 피드 기능 구현 (인기글, 구독글), Milestone 5: 프로필 관리 및 활동 점수 시스템, Milestone 6: 관리자 페이지로 총 6개의 마일스톤으로 나누었고, Milestone 1: 사용자 온보딩 및 인증 구현은 완료하였어. 다음으로 Milestone 2: 채팅방 및 실시간 대화 기능 구현을 개발하고 **자세한 UI 및 사용자 액션은 첨부파일을 검토하여 코딩을 작성해줘**

✦ Milestone 1: 사용자 온보딩 및 인증 구현을 위한 코딩 작성을 완료했습니다.

다음 파일들이 생성되었습니다:

\* src/supabaseClient.ts: Supabase 클라이언트 설정

\* src/StartScreen.tsx: 시작 화면 컴포넌트

\* src/EmailVerificationScreen.tsx: 이메일 인증 화면 컴포넌트

\* src/InterestSelectionScreen.tsx: 관심사 선택 화면 컴포넌트

\* src/NicknameSettingScreen.tsx: 닉네임 설정 화면 컴포넌트

\* src/pages/index.tsx: 온보딩 흐름을 조정하는 메인 페이지

\* src/pages/\_app.tsx: Next.js 앱의 커스텀 App 컴포넌트

\* src/styles/globals.css: Tailwind CSS 전역 스타일

\* package.json: 프로젝트 의존성 및 스크립트

\* tsconfig.json: TypeScript 설정

\* next-env.d.ts: Next.js 환경 타입 정의

\* postcss.config.js: PostCSS 설정

\* tailwind.config.js: Tailwind CSS 설정

Milestone 1 (사용자 온보딩 및 인증) 구현에 사용된 기술:

\* 프론트엔드 프레임워크: React (Next.js 프레임워크 내에서 사용)

\* 언어: TypeScript

\* 스타일링: Tailwind CSS

\* 백엔드 서비스 (인증 및 데이터베이스): Supabase (BaaS - Backend as a Service)

\* 배포 플랫폼 (예정): Vercel (Next.js 애플리케이션 배포에 최적화)

전체 프로젝트에 제안된 기술 스택:

\* 프론트엔드: React (TypeScript) with Next.js, UI 라이브러리 (예: Bootstrap, Material Design)

\* 백엔드: Node.js (Express.js, TypeScript) 또는 Python (FastAPI) (Supabase를 백엔드로 활용할 경우, 커스텀 백엔드 필요성은 줄어들 수 있습니다.)

\* 데이터베이스: Supabase (PostgreSQL 기반)

\* 실시간 통신: WebSocket (Supabase Realtime 기능 활용 가능)

\* AI/NLP: LLM API 연동 (예: OpenAI API)

\* 배포: Vercel

Milestone 2 작업 요약

1. 데이터베이스 스키마: 채팅 기능을 위한 chatrooms, messages, participants 테이블을 생성하고, 관련 보안 정책을 설정했습니다.

2. 채팅 페이지:

\* 사용자가 참여한 채팅방 목록을 볼 수 있는 페이지 (/chat)를 구현했습니다.

\* 실시간으로 메시지를 주고받을 수 있는 개별 채팅방 페이지 (/chat/[id])를 구현했습니다.

3. 실시간 기능: Supabase Realtime을 사용하여 새로운 메시지가 실시간으로 업데이트되도록 구현했습니다.

4. 채팅방 자동 참여: 사용자가 로그인하면, 프로필에 설정된 관심사를 기반으로 관련 채팅방에 자동으로 참여하는 기능을 구현했습니다.

-- Seed data for chatrooms

INSERT INTO chatrooms (name, description, interest) VALUES

('React', 'React, Next.js, and all things frontend.', 'Frontend'),

('Node.js', 'Discussing Node.js, Express, and backend development.', 'Backend'),

('Python', 'Everything about Python, from web dev to data science.', 'Python'),

('DevOps', 'CI/CD, Docker, Kubernetes, and more.', 'DevOps');