

무서핑 프로젝트 PRD (v0.8 개정판)

프로젝트 개요

무서핑은 공포 테마의 익명 커뮤니티로, 사용자가 AI로 생성된 귀신 캐릭터 이미지와 짧은 호러 설명을 공유하고 즐기는 웹 서비스입니다. 회원가입 없이 익명으로 게시글을 작성하고, 다른 이용자와 무서운 콘텐츠를 나누며, 사이드바의 미니게임 등 게이미피케이션 요소로 재미를 더합니다. 수익 모델은 Google AdSense 광고 게재이며, 모바일 웹 이용이 많을 것으로 예상되어 반응형 디자인과 모바일 최적화에 중점을 두었습니다. 초기 사용자 확보를 위해 유튜브 게임 스트리머 팬 커뮤니티 등에 노이즈 마케팅을 진행할 계획이며, 이용 시간대가 불규칙적으로 분산될 것을 고려해 서버리스 아키텍처로 트래픽 스파이크를 대응합니다. 성공적인 서비스 운영을 위해 익명성 유지, 콘텐츠 중재, 광고 정책 준수, 비용 관리 등의 도전 과제를 해결하는 것이 중요합니다.

- **타겟 사용자 및 차별점:** 호러 콘텐츠를 즐기는 20~30 대 웹 이용자들이 주요 타겟입니다. AI를 활용해 사용자가 손쉽게 공포 이미지를 생성하고 공유할 수 있다는 점이 기존 커뮤니티와 차별화됩니다. 별도의 앱 설치나 로그인 없이 웹에서 바로 즐길 수 있어 진입장벽이 낮고 익명성이 보장됩니다.
- **핵심 가치 제안:** “손끝으로 만드는 공포”라는 슬로건 하에, 누구나 AI가 그려준 귀신 이미지와 이야기를 통해 크리에이터가 될 수 있습니다. 또한 다른 사용자의 작품을 보며 무서움을 즐기고, 뽀빠이 터뜨리기 게임처럼 가벼운 놀이 요소로 긴장감을 풀 수 있습니다. 이러한 콘텐츠+엔터테인먼트 결합으로 사용자 체류 시간과 재방문율을 높이는 것을 목표로 합니다.
- **비즈니스 목표:** 서비스 출시 3개월 내 일일 활성 이용자(DAU) 1,000 명, 월간 게시글 500 개 이상을 달성하고, Google AdSense를 통해 서버 비용을 충당하는 것을 1차 목표로 합니다. 이후 트래픽 증가에 따라 대안 수익화 모델(예: 프리미엄 구독, 굿즈 연계 등)도 검토합니다. 다만 공포 이미지의 특성상 AdSense 충격적 콘텐츠 정책 위반 소지가 있어, 콘텐츠 필터링 강화와 정책 준수로 광고 계정 리스크를 최소화해야 합니다.

유저 플로우

무서핑 서비스에서 사용자가 경험하게 될 대표적인 유저 플로우는 다음과 같습니다:

1. **초기 진입**: 사용자는 커뮤니티 링크를 통해 무서핑 웹사이트에 방문합니다 (모바일/PC 웹 지원). 초기 랜딩 페이지는 최신 공포 게시글 피드로, **섬네일 이미지와 짧은 설명**이 카드 형태로 나열됩니다. 헤더나 첫 섹션에 간략한 서비스 소개와 “AI로 공포 만들기” 등의 **콜투액션(CTA)** 버튼을 배치해 신규 사용자의 이해를 돕습니다.
2. **게시물 열람**: 사용자는 관심가는 게시글을 클릭하여 상세 내용을 확인합니다. 게시글 상세 화면에는 **AI 생성 귀신 이미지, 짧은 호러 텍스트 설명**, 작성 시간, 좋아요 수 등이 표시됩니다. 이용자는 콘텐츠를 스크롤하며 **광고 배너**(예: 본문 상단 또는 중간에 표시)를 접할 수 있지만, 광고는 “스폰서” 레이블과 함께 콘텐츠와 시각적으로 구분되어 배치됩니다.
 - 만약 **부적절하거나 불쾌한 게시물**을 접할 경우, 사용자는 각 게시물에 제공되는 “**신고**” 아이콘을 눌러 관리자에게 해당 게시물을 신고할 수 있습니다. 신고 사유는 선택 또는 간단 입력으로 제출되며, 누적 신고가 일정 기준을 넘으면 해당 게시물은 **임시 블라인드** 처리되고 관리자가 검토합니다.
3. **게시글 작성 (AI 콘텐츠 생성)**: 사용자는 새로운 호러 게시물을 작성하고자 할 때, 화면의 “+ 생성하기” 버튼을 누릅니다. 별도의 작성 페이지 또는 모달이 열리며, 여기서 **AI 이미지 생성 및 텍스트 생성** 기능을 이용하게 됩니다.
 - **프롬프트 입력**: 기본적으로 별도의 회원 정보 입력 없이 바로 AI 생성 과정을 시작합니다. 사용자는 원하는 귀신 또는 배경에 대한 **간단한 문구를 입력**하거나, 제공되는 예시 키워드 중 하나를 선택할 수 있습니다 (예: “버려진 폐교의 원혼”, “피에 젖은 신부 귀신”).
 - **AI 이미지 & 텍스트 생성**: 사용자가 프롬프트를 확인하고 “생성”을 누르면, 백엔드 서버가 OpenAI의 DALL·E 이미지 생성 API와 GPT-3.5 텍스트 생성 API를 호출합니다. 약 **5~8초 정도의 대기 시간** 동안 로딩 스피너와 함께 “AI가 어둠 속에서 형상을 그리는 중...” 같은 메시지를 표시하여 사용자가 기다릴 수 있도록 합니다. 생성이 완료되면 **귀신 캐릭터 이미지와 그에 어울리는 무섭고 짧은 설명 글**이 미리보기 형태로 사용자에게 제시됩니다.
 - **콘텐츠 보정**: 사용자는 생성된 결과가 마음에 들지 않을 경우 “**다시 생성**” 기능을 통해 이미지를 1회 교체하거나, “**강화**” 기능으로 현 캐릭터를 약간 변형/업그레이드하는 연출을 시도할 수 있습니다. 강화 기능은 예를 들어 번개 이펙트와 함께 이미지의 일부 요소를 변화시키고 텍스트를 약간 수정하는 것으로, **최대 1~2회**로 제한됩니다 (남용 방지 차원).
 - **게시 및 공유**: 결과에 만족하면 사용자는 게시물 제목을 입력하거나 (제목을 생략하면 기본 값으로 “이름 없는 원혼” 등 자동 부여) “게시하기” 버튼을 누릅니다. 그러면 생성된 이미지와 텍스트가

하나의 새 게시글로 데이터베이스(Supabase)에 저장되고, 피드 상단에 노출됩니다. 이때 **작성 완료 후 3 분간**은 사용자의 브라우저 세션에 편집 토큰이 유지되어, 해당 게시글 화면에서 “수정” 또는 “삭제” 버튼을 통해 내용을 고치거나 삭제할 수 있습니다. (3 분 초과 후에는 일반 사용자는 수정/삭제 불가)

- **익명성 보장**: 게시물에는 작성자의 닉네임이나 ID 가 일절 표시되지 않으며, **익명 아이콘**이나 “떠도는 혼령” 같은 표현으로 표기됩니다. 작성자는 로그인 없이도 JWT 기반 편집 토큰으로 본인 게시글을 잠시 관리할 수 있지만, 인증 정보는 브라우저 쿠키에 HttpOnly 로 저장되어 노출을 최소화합니다. (OpenAI API 호출 역시 클라이언트에서 직접 이뤄지지 않고, 안전을 위해 백엔드 서버를 경유합니다.)

4. **좋아요 및 월간 베스트**: 게시글 상세 화면에서 마음에 드는 콘텐츠라면 **하트 대신 피 눈물 모양**으로 커스텀 디자인된 “좋아요” 버튼을 누를 수 있습니다. 한 사용자가 같은 게시물을 여러 번 좋아요할 수 없도록 **쿠키 및 세션 확인, IP 기반 제한**을 적용합니다. 좋아요를 누르면 **애니메이션 효과**(피가 떨어지는 이펙트 등)와 함께 즉시 좋아요 수가 증가하며, 사용자의 UX 를 해치지 않도록 Optimistic UI 를 사용하여 백엔드 응답 전에 미리 UI 에 반영합니다. 좋아요 수는 추후 **인기순 정렬 및 월간 베스트 선정**에 활용됩니다.

- 메인 피드 화면에서 사용자는 상단 필터를 통해 “최신순”, **“인기순”**으로 게시글 정렬을 전환할 수 있습니다. 인기순은 일정 수 이상의 게시글이 쌓이면 **최근 24 시간 내 가장 좋아요가 많은 순 또는 전체 좋아요 순** 등으로 정의됩니다 (구체적 알고리즘은 지속 튜닝).
- 매월 말일을 기준으로 해당 월에 **좋아요를 가장 많이 받은 상위 N 개 게시글**을 선정하여, **월간 베스트 페이지**에 노출합니다. 이 페이지는 다음 달 1 일에 자동 생성되며, 사용자들은 지난달의 화제작을 한눈에 볼 수 있습니다. 월간 베스트 코너는 커뮤니티의 **Hall of Fame** 역할을 하여, 양질의 콘텐츠 생산을 독려합니다.

5. **부가 기능 (미니게임)**: 사용자가 콘텐츠를 소비하다가 잠시 쉴 수 있도록, 화면 사이드바에 “**뽁뽁이 터뜨리기**” 미니게임이 제공됩니다. 사용자는 사이드에 배치된 뽁뽁이 비닐 UI 를 터치/클릭하여 **톡톡 터뜨리는 소리**를 들으며 간단한 재미를 느낄 수 있습니다. 연속으로 터뜨린 개수에 따라 세션 누적 **점수**가 표시되고 작은 아이콘 애니메이션 (예: ‘+1’ 효과)이 나타납니다. 이 점수는 사용자의 기기(세션)에 임시 저장되며, 새로고침 시 리셋됩니다. 점수는 랭킹 등에 사용되지는 않고 **개인 기록용**이지만, 차후 로그인 시스템 도입 시 **포인트나 배지**로 전환을 검토할 수 있습니다.
6. **종료 및 재방문 유도**: 사용자가 세션을 종료하려고 할 때, 예를 들어 일정 시간 이상 활동한 사용자에게 대해서는 “무서핑을 즐겨주셔서 감사합니다! 다음에 더 오싹한 이야기로 만나요 🐼” 등의 **마무리 메시지**를 띄워 긍정적인 인상을 남깁니다. 게시글 공유 기능을 통해 트위터, 페이스북 등 SNS 로 퍼갈 수 있게 하여 **자발적 바이럴**을 기대합니다. 즐겨찾기 또는 홈 화면

바로가기(PWA 기능)를 통해 재접속이 편리하도록 유도하고, **신규 콘텐츠 알림** 또는 **이벤트 알림** (예: “이달의 귀신 콘테스트 개최”)을 추후 Push 로 제공할 계획입니다.

핵심 기능

무서핑 서비스의 핵심 기능과 각 기능별 사양, 고려 사항은 다음과 같습니다:

- **익명 게시판:** 회원가입이나 로그인 절차 없이 **익명으로 글을 작성**하고 조회할 수 있는 게시판입니다. 각 사용자는 방문 시 브라우저 세션당 무작위 **UUID 식별자**가 할당되어, 별도 인증 없이도 일정 기간(예: 24 시간) 동안 동일인으로 간주됩니다. 이를 통해 **동일 세션 내에서 본인이 작성한 글을 편집/삭제**하거나 좋아요 중복 방지를 처리합니다. 익명성을 유지하면서도 편집 권한을 부여하기 위해, 게시글 작성 시 **편집 JWT 토큰**을 발급하여 응답에 포함시키고, 클라이언트는 이를 HttpOnly 쿠키로 저장합니다. 게시 후 **3분 이내** 해당 토큰을 제출하면 수정/삭제를 할 수 있으며, 시간이 지나거나 쿠키를 지워 토큰이 없으면 이후엔 관리자만 삭제 가능합니다.
 - **보안:** 익명 게시판 특성상 **쿠키 삭제, 시크릿 모드 사용, VPN 변경** 등으로 하나의 사용자가 다수의 세션을 획득해 **중복 투표나 도배**를 시도할 수 있습니다. 이를 보완하기 위해 **IP 기반 레이트 리미팅**을 도입하여 일정 시간 내 과도한 요청(예: 게시글 연속 등록, 좋아요 폭주)을 차단합니다. 또한 Cloudflare Turnstile 같은 **봇 감지 CAPTCHA**를 회원가입 대신 **게시글 작성 액션에 적용**하여, 자동화된 스팸을 예방합니다. 세션 쿠키에는 HttpOnly, Secure, SameSite 옵션을 적용하고, 중요한 API 엔드포인트에는 **CSRF 토큰 검증**을 추가하여 **악성 스크립트 공격**을 방지합니다.
- **AI 콘텐츠 생성 (이미지 & 텍스트):** 무서핑의 차별화된 핵심 기능으로, OpenAI의 **DALL·E API**를 통해 귀신 이미지를 자동 생성하고, **GPT-3.5 Turbo**를 통해 해당 이미지를 설명하는 짤막한 호러 스토리를 생성합니다. 모든 AI API 호출은 **백엔드 서버리스 함수**를 통해 이뤄지며, API Key는 공개되지 않도록 안전하게 서버 환경변수로 관리합니다 (OpenAI 권고사항: “API 키를 브라우저나 앱에 노출하지 말고 반드시 백엔드를 경유할 것”).
 - **이미지 생성:** 기본 해상도 1024×1024의 이미지를 사용하며, OpenAI 가격 기준 **1회당 \$0.02**의 비용이 발생합니다. 생성된 원본 이미지는 **Vercel Blob 스토리지**나 **Supabase 스토리지**에 업로드되어 영구 보관됩니다. 이미지는 Next.js의 **Image 컴포넌트**를 통해 자동 **웹 최적화(.webp 변환, 사이즈 조절)** 후


CDN 캐시를 거쳐 제공되어, 사용자 단말에서 빠르게 로드됩니다. 생성 시간은 보통 5 초 내외이지만, 간혹 10 초 이상 지연 시 **타임아웃 처리 및 재시도** 로직이 백엔드에 구현되어 있습니다. 사용자는 생성 대기 중 취소하거나, 너무 오래 걸릴 경우 결과 대신 실패 메시지를 받게 됩니다.

- **텍스트 생성**: 이미지 생성과 병렬로 OpenAI **GPT-3.5 Turbo** 모델에 프롬프트(예: “다음 묘사를 50 자 내외 한국어 공포 분위기로 써줘: [이미지 설명]”)를 보내, 짧은 호러 텍스트를 생성합니다. 비용은 토큰 사용량에 따라 **1,000 토큰당 \$0.002** 수준으로 매우 저렴하지만, 길이를 2~3 문장으로 제한하여 호출 비용을 최소화합니다. 추후 OpenAI 의 차세대 모델이나 Google 의 **Gemini API** 출시 시 품질 및 비용에 따라 전환을 검토하고 있습니다. 이를 위해 AI 호출 로직을 **Adapter 패턴**으로 구현하여, OpenAI 또는 Google API 모듈을 쉽게 교체할 수 있도록 해두었습니다.
- **NSFW 및 부적절 내용 필터**: AI 생성 결과가 서비스 정책에 어긋나지 않도록 **다중 필터링 단계**를 거칩니다. 우선 OpenAI 의 API 자체에서 **콘텐츠 필터**를 적용하며, 추가로 OpenAI 의 **Moderation API** 를 활용해 생성된 텍스트의 유해성을 점검합니다. OpenAI Moderation 최신 모델(GPT-4o 기반)은 **40 개 언어에서 95% 정확도로 유해한 텍스트를 감지**하는 강력한 도구이며, 개발자에게 무료로 제공됩니다 . 텍스트 생성 결과가 욕설/증오/폭력적인 경우 **flag** 처리하여 사용자에게 출력하지 않고 폐기합니다. 이미지의 경우 OpenAI 가 노출이 심한 이미지 생성을 차단하지만, 추가로 **Google Cloud Vision SafeSearch API** 등을 활용해 폭력성(피, 시체 등)이나 선정성 여부를 검사하고, 부적합 판정 시 해당 이미지를 업로드하지 않습니다. 이러한 다중 필터링으로 **AdSense 정책** 및 **청소년 보호법** 위반 소지를 사전에 차단합니다.
- **게시물 피드 및 정렬**: 메인 화면에는 최신 게시물이 작성 시간 역순으로 정렬되어 무한 스크롤 형태로 표시됩니다. 각 게시물 카드에는 썸네일 이미지, 짧은 설명, 좋아요 수, 작성 시각 등이 보입니다. 사용자는 상단 메뉴에서 **정렬 옵션**을 선택해 인기순 보기로 전환 가능하며, 인기순은 기본적으로 최근 24 시간 기준으로 좋아요 수가 많은 순입니다 (시간 범위와 가중치는 추후 조정). 이때 인기 순위를 효율적으로 계산하기 위해 **Redis** 캐시를 사용합니다. 게시물 작성이나 좋아요 발생 시 **Supabase DB** 와 **Upstash Redis** 에 동시에 카운트를 기록해두고, 조회 시에는 Redis 의 집계 데이터를 활용하여 속도를 높입니다. 데이터 정합성은 일정 주기마다 백엔드 작업(크론 잡)으로 Redis 수치를 DB 와 동기화하여 맞춥니다.
 - 또한 매월 최고의 인기를 얻은 게시물들을 모아 **월간 베스트 페이지**를 운영합니다. 월간 베스트는 매월 1 일 자동 생성되며, 전월 동안 좋아요 최다 순 상위 10 개 내외의 게시물이 카드 형식으로 큼직하게 전시됩니다. 이 페이지는 Next.js 의 **ISR(Incremental Static Regeneration)** 기능을 사용해 한 번 생성 후 정해진 기간

캐싱함으로써, 많은 사용자가 접속해도 DB 부하 없이 빠르게 제공됩니다.

- **좋아요 및 리액션:** 무서핑만의 정체성을 담은 커스텀 좋아요 아이콘을 통해 사용자 반응을 유도합니다. 기본 아이콘은 빨간 **피 방울 모양**으로, 클릭 시 하트 모양으로 터지는 애니메이션과 함께 카운트가 증가합니다. 한 사용자가 동일 게시물을 여러 번 좋아요할 수 없도록 **브라우저당 1 회로 제한**하며, 이를 위해 첫 클릭 시 로컬 스토리지나 쿠키에 해당 게시물 ID 를 저장합니다. 추가로 서버에서도 요청 시 JWT 또는 세션 ID 와 IP 를 검사하여, **동일 IP 대역에서 과도한 좋아요 발생** 시 차단합니다. (예: 1 분 내 동일 IP 에서 10 회 이상 좋아요 요청이면 Rate Limit 적용)

좋아요 수는 게시물 목록에서 **인기 지표**로 노출되며, 일정 수 이상 누적되면 리스트 카드에 🔥 이모지 등을 함께 표시해 눈에 띄게 합니다. 추후 **리액션** 기능을 확대하여, “소름 돋아😱”, “웃겨요😂” 같은 여러 반응으로 다양하게 표현할 수 있도록 업그레이드 가능성을 열어둡니다.

- **광고 수익화 (Google AdSense):** 서비스 운영 비용을 감당하기 위해 **구글 애드센스 광고**를 페이지 내에 통합합니다. 광고 배치는 사용자 경험을 크게 해치지 않는 선에서, **반응형 디스플레이 광고** 위주로 넣을 계획입니다. 예를 들어, **헤더 상단**에 가로형 배너 1 개, **사이드바 상단**에 고정 배너 1 개, 그리고 **피드 목록 중간중간** 및 **게시글 상세 하단** 등에 네이티브 광고 형식으로 삽입합니다. 각 광고 영역에는 “Ad” 또는 “광고” 라벨을 명시하여 콘텐츠와 착각하지 않도록 하고, 이미지/텍스트 콘텐츠와 **최소 150px 이상** 이격되도록 CSS 디자인을 조정합니다.
 - *AdSense 정책 고려:* 공포 컨셉트상 **잔혹한 이미지나 선정적 요소**가 광고와 함께 노출될 경우, Google 정책 위반으로 계정이 제한될 수 있습니다. 실제 AdSense 정책에서는 “폭력적인 언어나 잔혹하거나 역겨운 이미지(피, 내장, 시체 등)가 포함된 콘텐츠”에 광고 게재를 허용하지 않습니다. 따라서 **콘텐츠 중재 시스템**을 통해 이러한 이미지/텍스트가 서비스에 올라오지 않도록 해야 하며, 혹시 발생 시 신속히 제거해야 합니다. 또한 **AI 생성 콘텐츠 표시 의무**에도 대비해, 필요 시 “ AI 생성” 라벨을 게시물에 표기하는 방안도 고려합니다 (일부 광고주는 AI 콘텐츠 미표시를 문제 삼을 수 있음).
 - *수익 및 차단 대응:* 광고 노출/클릭으로 월 \$XX 수익을 예상하지만, **광고 차단기(AdBlock)** 사용자가 많을 경우 수익이 감소할 수 있습니다. 이를 위해 광고 스크립트 로드 여부를 감지하여, 차단된 자리에는 “**광고로 무료 서비스가 유지됩니다**” 등의 안내를 띄우는 방안을 고려합니다. 사용자가 계속 AdBlock 을 사용할 경우 강제 대응은 하지 않고, 대신 추후 **프리미엄 구독** 도입 시 광고 없이 이용 옵션을 제공하는 방향으로 검토합니다. 또한 **부정 클릭**(예: 동일 사용자가 반복 클릭) 모니터링을 통해 Google 무효 트래픽 정책 위반이 없도록 유의합니다.

- 콘텐츠 모더레이션 및 안전 장치:** 익명 커뮤니티 특성상 악성 콘텐츠 업로드와 이용자 간 분쟁 가능성이 높으므로, 서비스 론칭 단계에서 강력한 모더레이션 정책을 시행합니다. 우선 **욕설 및 비속어 필터링**을 게시물 제목 및 AI 프롬프트 입력 단계에 적용합니다. 오픈소스 한글 비속어 리스트인 **badwords-ko** 라이브러리를 활용하고, 자음/모음 분리나 특수문자 교체 형태의 변형 욕설도 정규식으로 탐지합니다. 예를 들어 “ㅅㅂ” 또는 “욕@@” 등의 패턴도 최대한 걸러내며, 사용자가 이를 입력하면 안내 메시지와 함께 해당 단어를 제거하거나 게시를 막습니다.
 - AI 악용 방지:** OpenAI 이용 약관에 따라, 사용자 프롬프트에 노골적인 폭력, 성적 내용이 들어가면 AI 이미지가 생성되지 않도록 백엔드에서 **차단 키워드**를 검사합니다. (예: “피가 낭자한”이라는 프롬프트는 DALL·E가 거부할 수 있으므로 미리 걸러내거나 완곡한 표현으로 수정 안내) 또한 사용자들이 우회적으로 금지 콘텐츠를 생성하려는 시도를 파악하기 위해 **로그 모니터링**을 시행합니다. AI 모델이 생성한 텍스트 역시 OpenAI Moderation을 통해 성희롱, 혐오 발언 등을 걸러내며, 한국어의 경우 OpenAI 필터가 놓치는 부분은 **사람이 수동 검토**할 수 있도록 로그에 남깁니다.
 - 신고 및 관리자 대응:** 커뮤니티 신뢰도 유지를 위해 **사용자 신고** 기능을 제공합니다. 각 게시물에는 신고 버튼이 있고, 누적 신고가 많거나 명백히 정책 위반인 경우 **자동 블락** 처리합니다. 운영자는 별도의 **관리자 대시보드**에서 블락된 게시물을 확인하고 최종 삭제 또는 복구 결정을 내립니다. 악성 유저에 대해서는 서버에 저장된 **IP 해시값**을 기반으로 일정 기간 차단할 수 있습니다. (IP 자체는 개인정보이므로 해시+솔트로 저장하고, 법적 문제 소지가 있는 경우에 한해 원시 IP를 복원해 수사협조 가능) 이때 차단된 사용자가 VPN 등으로 우회할 가능성도 있어, Cloudflare 등 네트워크 레벨의 방어도 함께 고려합니다.
 - 법적 준수:** 서비스가 **청소년 유해매체물**로 간주되지 않도록 특히 신경 씁니다. 일단 **19금 성인물이나 지나친 잔혹물은 생성 금지**하며, 전체이용가 수준의 공포만 취급합니다. 부득이 일부 센스 있는 수위의 콘텐츠가 나오더라도 게시물 앞부분에 “**잔혹한 표현 주의**” 등의 경고 문구를 추가할 수 있습니다. 또한 **개인정보 보호**를 위해 이용자의 어떠한 신상 정보도 수집하지 않으며, 필요한 최소한의 데이터(예: IP 해시, 쿠키 식별자, 기기 정보 정도)만 활용합니다. 서비스 이용 시 쿠키가 사용되므로 **쿠키 동의 배너**를 통해 사용자에게 알리고 동의를 받습니다. 서버에 저장된 로그/데이터 중 IP 등 잠재적 개인정보 요소는 **최대 1년 이내**로 주기적으로 파기하여, 개인정보보호법 준수 및 이용자 프라이버시 보호에 힘씁니다.
- 유저 등급/포인트 시스템 (향후 제안):** 현재는 완전 익명 서비스로 별도의 유저 프로필이나 등급이 없습니다. 그러나 커뮤니티 활성화 및 건전성 유지를 위해 추후 **유저 등급 제도**를 도입하는 것을 검토할 수 있습니다. 예를 들어, 일정 이상 건전한 활동을 한 사용자에게 **호러 테마 배지**를

부여하거나, AI 콘텐츠 생성 횟수를 추가로 부여하는 **레벨 업 보상 시스템**을 마련할 수 있습니다. 또는 프리미엄 구독 유저에게 **특별 테마 제공**이나 **광고 미노출** 등의 혜택을 줄 수도 있습니다. 다만 익명성을 해치지 않는 선에서 이루어져야 하므로, 로그인 시스템 도입 시 최소한의 프로필 정보만 받고 등급을 운영하거나, 현 시점에서는 **세션 단위의 임시 포인트**로 가볍게 적용하는 방안을 고려합니다. 유저 등급화는 참여 동기를 높이고 책임감을 부여하여 양질의 콘텐츠 생산과 커뮤니티 **자정 작용**을 높일 것으로 기대됩니다.

기술 스택

무서핑 서비스는 **모던 웹 기술 스택**을 기반으로 개발되며, 서버리스 아키텍처와 제 3 자 API 를 적극 활용합니다. 주요 기술 구성은 다음과 같습니다:

- **프론트엔드**: React 기반의 **Next.js 14** 를 사용합니다. 페이지는 대부분 SSR/SSG 로 구축하여 초기 로딩을 빠르게 하고 SEO 에 유리하도록 했습니다. 스타일링은 **Tailwind CSS** 로 커스텀 디자인을 구현하고, 일부 컴포넌트에는 **Chakra UI** 를 도입해 접근성과 반응형 디자인 편의성을 높였습니다. 데이터 페칭과 상태 관리는 ****React Query(SWR)****로 처리하여, 예를 들어 게시글 목록이나 좋아요 상태를 **캐싱**하고 **실시간 업데이트(revalidate)**합니다. 좋아요 클릭 시 **Optimistic UI** 업데이트로 **즉각 반영 후 백엔드 검증**하는 패턴을 적용했습니다.
 - **이미지 최적화**: Next.js 의 <Image> 컴포넌트를 사용해 자동으로 이미지 크기 최적화, WebP 포맷 변환, Lazy Loading 을 적용했습니다. 덕분에 AI 생성 이미지도 사용자의 기기에 맞게 리사이즈되어 전송되고, 화면에 보일 때만 로드되어 **트래픽 절감**과 **렌더링 성능 개선** 효과를 얻습니다. 전세계 이용자에게 빠른 응답을 제공하기 위해 Vercel 의 글로벌 CDN 이 이미지 정적 자산을 캐시하며, 필요 시 Cloudflare CDN 과 연계도 고려 중입니다.
 - **UI/UX**: 다크 테마 기반의 UI 로 공포 분위기를 살리되, 텍스트 가독성을 위해 적절한 대비를 유지했습니다. 모바일 퍼스트 디자인 원칙으로 작은 화면에서도 레이아웃이 깨지지 않게 하고, **터치 이벤트** 최적화 및 **모바일 폰트 사이즈** 조정을 했습니다. PC 웹에서는 여백과 사이드바를 활용해 한 화면에 많은 정보가 보이도록 했습니다. 또한 사용자 편의를 위해 주요 인터랙션(게시물 생성, 좋아요 등)에 **사운드 효과**(예: 카메라 셔터음, 심장 소리)를 옵션으로 제공하여 몰입감을 더했습니다.
 - **모니터링**: 프론트엔드 오류나 성능 추적을 위해 **Sentry** 를 연동하여 JS 예외 발생 시 개발팀에 리포팅합니다. 사용자 행동 분석은

가볍게 Google Analytics 4 를 삽입하여 페이지뷰, 체류 시간 등을 집계하지만, **개인식별정보(IP, 쿠키 등)**는 수집하지 않도록 최소한으로 활용합니다.

- **백엔드: Vercel Serverless Functions** (Node.js 환경)로 구현됩니다. Next.js 의 API Routes 를 활용하여 /api/* 경로로 각종 기능(게시글 CRUD, AI 호출, 좋아요 처리 등)을 제공합니다. 서버리스 함수는 요청시에만 실행되며, 무상태이므로 **DB 접근**이 필요할 때마다 Supabase 클라이언트를 통해 연결합니다. 중요한 API (예: 메인 피드 리스트) 일부는 Vercel **Edge Function** 으로 배포하여 지리적으로 사용자와 가까운 곳에서 응답하도록 하여 레이턴시를 최소화했습니다. 다만 서버리스 특성상 **콜드 스타트** 지연이 발생할 수 있어, 배포 시 함수 번들 크기를 줄이고, 지역 리전을 서울 등으로 지정하여 cold start 를 줄이도록 최적화했습니다. (추가로 Vercel 의 bundle analyzer 와 **experimental-edge** 옵션을 사용해 함수 초기 구동을 튜닝 중입니다.)
 - **데이터베이스: **Supabase (PostgreSQL)****를 메인 DB 로 사용합니다. Supabase 를 선택한 이유는 PostgreSQL 의 안정성과 확장성에 더해, 인증/스토리지 등 부가 기능과 손쉬운 REST API 제공 등 개발 편의성이 높기 때문입니다 . 게시글 테이블, 좋아요 테이블 등에 적절한 **인덱스**를 추가하여 최신순, 좋아요순 쿼리의 성능을 확보했습니다. 예를 들어 posts 테이블의 created_at 필드와 like_count 필드에 인덱스를 생성하여 정렬 시 빠르게 동작하게 했습니다. 또한 Supabase 의 ****Row Level Security (RLS)****를 활용, 게시글 수정/삭제 API 에서는 요청에 포함된 JWT 의 게시자 식별자와 테이블의 작성자 필드가 매칭될 때만 허용되도록 정책을 설정했습니다 . 이를 통해 혹시 토큰이 탈취되거나 변조되는 경우를 방지하고자 합니다.
 - **캐싱 및 실시간:** 일부분 데이터는 **Upstash Redis**(Vercel KV)를 도입하여 캐시합니다. 예를 들어 인기 게시글 리스트, 전체 좋아요 수 집계 등은 Redis 에 키-값으로 저장해두고 빠르게 조회합니다. Redis 는 eventual consistency 모델이므로 약간의 일치 지연은 감안해야 하지만, 일정 주기로 DB 와 싱크를 맞춰 큰 차이는 없도록 유지합니다. 또한 추후 사용자 수 증가 시 **Supabase Realtime** 기능을 활용하여, 새로운 게시물이 올라오면 피드에 실시간으로 추가하거나, 좋아요 증가가 WebSocket 으로 브로드캐스트되어 모두의 화면에 동시에 업데이트되는 기능도 고려하고 있습니다.
 - **파일 스토리지:** AI 생성 이미지를 비롯한 사용자 콘텐츠 이미지는 영구 저장이 필요합니다. 현재는 Vercel 의 **Blob Storage** 를 사용하여 업로드/호스팅하고, 기본 제공되는 Edge CDN 을 통해 전송합니다. Blob Storage 는 사용이 간편하지만 벤더 종속적이므로, 향후 트래픽이나 비용을 고려해 **Supabase Storage** 로 이전하거나 병행 사용할 계획도 있습니다. 두 스토리지 모두 CDN 경로 제공 및 이미지 최적화 기능이 있어, 사용자 다운로드 속도에는 문제가 없습니다. 동영상 등의 미디어는 취급하지 않으므로 이미지 위주로

저장 용량을 산정하고 있으며, 초기에는 월 5GB 내외로 가볍게 시작해 추후 수십~수백 GB 까지 확장할 수 있도록 예산을 책정합니다.

- 외부 API 및 서비스:

- **OpenAI API**: 앞서 설명한 대로 이미지 생성용 DALL·E 와 텍스트 생성용 GPT-3.5 Turbo 모델을 사용합니다. OpenAI API 호출은 서버 측에서 이뤄지며, API 키는 Vercel Dashboard 의 환경 변수로 저장되어 노출을 방지합니다. OpenAI Best Practice 에 따라 브라우저에는 절대 키를 노출하지 않고 백엔드 경유를 철저히 지킵니다 . 월 API 예산은 초기 약 \$20 정도로 책정하고, 지속적으로 **Usage 모니터링**을 통해 비용을 추적합니다. 만약 이용량 증가로 비용 급등 시 **이미지 생성 횟수 제한**(예: 비회원 일 5 회)이나 **저해상도 무료+고해상도 유료** 전략, 또는 **오픈소스 대안 모델 활용** 등을 검토합니다. 추후 Google 의 **Imagen**(이미지) API 나 **PaLM/Gemini**(텍스트) API 가 안정되면 품질과 가격 경쟁력을 비교 평가하여 도입을 검토할 것입니다.
- **Google AdSense**: 광고 삽입을 위해 Next.js 에 **next/script** 태그로 AdSense 코드를 추가하고, 광고 위치마다 고유 슬롯 ID 를 설정합니다. Ads 는 **클라이언트 사이드**에서 로드되므로 초기 SSR 에는 포함되지 않고, 브라우저가 실행되면서 광고 스크립트가 필요한 공간을 채우는 방식입니다. 이 동적 로드의 단점은 콘텐츠 레이아웃 시프트(밀림) 현상이 생길 수 있다는 것인데, 이를 막기 위해 광고 자리 크기를 미리 CSS 로 예약하고, 로딩 중에는 회색 배경 등으로 placeholder 를 줍니다. 또한 Intersection Observer 로 화면 진입시에만 광고 로드를 트리거하여, 사용자가 보지도 않을 광고를 미리 불러와 리소스를 낭비하지 않도록 했습니다.
- **콘텐츠 모더레이션 도구**: OpenAI Moderation API(텍스트) 외에도, **Google Perspective API** 를 도입 고려 중입니다. Perspective API 는 댓글 등 UGC 의 ****악성도(Hate, Toxicity)****를 0~1 사이 점수로 평가해주는 도구로, 특히 한국어에 대해서도 학습되어 있어 도움이 될 수 있습니다. 다만 OpenAI Moderation 으로 상당 부분 커버 가능하기에 우선은 도입을 유보하되, 문제 발생 시 보조적으로 사용합니다. 이미지 모더레이션은 앞서 언급한 Cloud Vision SafeSearch 외에, 필요하면 **Api4.ai** 나 **NSFW.js** 등의 라이브러리로 추가 검증을 합니다. 예를 들어, 매우 드문 케이스지만 AI 가 생성한 이미지에 사람이 **실제 연예인 얼굴**과 흡사하게 나오는 경우 퍼셉트론 알고리즘으로 판별해 차단하는 등의 조치를 취할 수 있습니다.
- **기타**: 서비스 안정성을 위해 **Cloudflare** 를 프록시로 세워 DDoS 공격이나 봇 크롤링을 억제합니다. Cloudflare 의 Bot Management 기능이 사용자의 이상 행동을 탐지하면, 해당 요청에 대해 CAPTCHA 를 추가로 요구하거나 차단합니다. 또한 데이터 백업을

위해 정기적으로 Supabase DB 의 백업 덤프를 별도 스토리지(S3 등)에 저장하고, 장애 시 복구 절차를 마련합니다.

- **성능 및 확장성:** 초기 사용자 규모에서는 서버리스+단일 DB 조합으로 충분히 감당 가능하나, **사용자 급증 시 대응 방안**도 준비해두고 있습니다.
 - DB 부하를 줄이기 위해 ****읽기 전용 복제본(Read Replica)****을 활용, SELECT 쿼리는 복제본으로 분산하고 메인 DB 는 쓰기 전용으로 부담을 낮춥니다. 또한 **PgBouncer** 등 커넥션 풀러를 사용해 서버리스 함수 호출 시 매번 새로운 DB 세션을 생성하는 문제를 완화합니다 (Supabase 에서 제공되는 **프 connection** 기능 활용).
 - 서버리스 함수는 Vercel 환경에서 자동 확장되지만, 일정 시간 미사용 시 인스턴스가 종료되어 **콜드 스타트** 지연이 발생합니다. 이를 완화하고자 자주 쓰는 함수는 주기적으로 워머(ping)를 넣거나, 한 함수에 여러 기능을 통합하여 호출 빈도를 높임으로써 재활용되도록 했습니다. 또, Webpack 번들링 시 **필요한 모듈만 포함**하여 함수 사이즈를 최소화하고, Vercel 의 **Edge Network** 에 함수를 배포하여 사용자와의 물리적 거리를 줄였습니다.
 - 프론트엔드는 정적 파일로 배포되므로, Vercel 이 자동으로 CDN 캐싱을 합니다. CSS, JS, 이미지 등은 요청 지역에 따라 가장 가까운 PoP(Point of Presence)에서 서빙되어 글로벌 성능을 최적화합니다. 필요시 핵심 페이지는 ****정적 생성 (SSG)****으로 전환해, 매 요청마다 서버 함수를 거치지 않고도 콘텐츠를 제공할 수 있게 할 계획입니다.

요약하면, 무서핑 프로젝트는 **Next.js + Supabase + OpenAI + Vercel** 이라는 최신 스택 조합을 활용하여 개발되고 있으며, 익명 커뮤니티 운영에 필요한 성능, 보안, 모니터링, 정책 준수 측면을 세심하게 설계하였습니다. 모든 OpenAI API 키는 서버에 안전하게 보관되고, **클라이언트에서는 절대 직접 호출하지 않는** 등 기본 보안 수칙을 준수합니다 . 또한 AdSense 와 같은 외부 서비스의 가이드라인(예: **충격적인 콘텐츠 금지**)을 엄격히 지키고, OpenAI 의 강력한 **콘텐츠 필터링 툴** 을 활용하여 서비스 신뢰성을 높입니다. 기술 스택 선정과 설계에 있어서는 **확장 가능성**을 열어두어, 향후 사용자가 늘어나거나 요구사항이 변경되더라도 유연하게 대응할 수 있도록 하였습니다.