

무서핑 웹 서비스 PRD

개요

무서핑은 React 기반으로 Vercel(Vercel v0) 위에 구축된 AI 기반 유머/공포 콘텐츠 생성 서비스입니다. 현재는 프론트엔드만 제공되며, 사용자는 AI가 생성한 귀신·캐릭터 콘텐츠를 감상할 수 있습니다. 다음 개발 단계에서는 공개 게시판 기능과 광고 수익화 기능을 추가하여 사용자 참여를 높이고 수익화를 추진합니다. 본 PRD는 게시판 기능의 상세 요구사항과 Google AdSense 기반 수익화 통합 방안을 규정하며, CI/CD 배포 및 운영·안전성 측면도 다룹니다.

사용자 및 인증

- 익명 게시판: 회원가입이나 로그인 없이 모든 사용자가 게시글 작성 및 조회, 댓글 작성, 좋아요 등을 할 수 있어야 합니다.
- 익명성 유지: 로그인 기능 없이 각 기능을 제공하되, 악용 방지를 위해 쿠키나 세션을 통한 간단한 중복 방지 로직(예: 동일 기기에서 반복 좋아요 제한)을 고려할 수 있습니다.
- 미래 확장성: 현재는 인증 모듈 없이 익명으로 처리하며, 추후 필요 시 소셜 로그인(예: OAuth) 등의 인증 도입 가능성을 열어둡니다.

게시판 기능

- 게시글 작성: 사용자는 익명으로 텍스트와 AI 생성 이미지를 포함한 게시글을 작성할 수 있어야 합니다. 게시글 작성 시 귀신 이름, 속성 등의 메타데이터와 선택적 태그를 입력할 수 있어야 합니다. 이미지 업로드 후에는 미리보기를 제공해 사용자 경험을 높입니다.
- 게시글 수정·삭제: 사용자가 작성 직후 일정 시간 동안만 수정이나 삭제할 수 있도록 합니다. 예를 들어 '게시 후 5분 이내 편집 허용'과 같은 방식입니다.
- 좋아요 기능: 각 게시글에 좋아요 버튼을 달아 사용자가 공감 표시를 누를 수 있도록 합니다. 로그인 없이 단순 카운터로 처리하되, 중복 클릭 방지를 위해 쿠키나 IP 기반 단순 필터링을 구현할 수 있습니다.

- 정렬 기능: 게시글 목록을 최신순(작성 시간 기준)과 인기순(좋아요 수 기준)으로 정렬할 수 있어야 합니다. 인기순 정렬은 사용자 참여를 유도하는 데 유용합니다.
- 댓글 시스템: 모든 사용자가 익명으로 댓글을 남길 수 있도록 합니다. 댓글은 프로그래밍 방식의 기본 욕설 필터링을 적용해 부적절한 언어를 자동으로 차단하거나 대체합니다. 예를 들어 JavaScript용 욕설 필터 라이브러리(bad-words, 한국어용 badwords-ko 등)를 활용할 수 있습니다 [OBJ].
- 데이터 저장소(백엔드): 프론트엔드 중심 서비스이므로 서버리스 저장소를 이용합니다. 예를 들어 Vercel KV(Upstash Redis 기반), Supabase(Postgres), Firebase Realtime Database/Firestore 등이 적합합니다. Supabase는 Postgres 기반으로 브라우저 기반 편집기와 인증/저장소 기능을 제공하며 빠른 개발을 지원합니다 [OBJ]. Upstash Redis(Vercel KV)도 서버리스.엣지 환경에 최적화되어 빠르게 인스턴스를 띄울 수 있습니다 [OBJ] [OBJ].
- 미리보기 및 UI: 이미지 파일 선택 시 즉시 미리보기를 보여주고, 게시글 내용은 간단한 마크다운 또는 텍스트로 표시합니다. UI 라이브러리로는 React 기반의 TailwindCSS 또는 Chakra UI 등을 사용해 가독성을 높일 수 있습니다.

광고 수익화

- 배너 광고 배치: 구글 AdSense를 이용해 페이지 상단(헤더), 사이드바, 게시글 목록 사이(피드), 각 게시글 내 콘텐츠 중간(paragraph 사이) 등에 배너 광고를 배치합니다. 특히 긴 글일 경우 본문 중간(paragraph 사이)에 광고를 삽입하면 높은 성과를 기대할 수 있습니다 [OBJ]. 또한 사이드바에 고정형 스크롤 배너를 두면 지속적으로 노출되므로 수익 개선에 도움이 됩니다 [OBJ].
- 애드센스 단위: 데스크톱 및 모바일 반응형(adaptive) 광고 단위를 사용하여 다양한 화면 크기에 대응합니다. 네이티브 애드(콘텐츠와 유사한 형식의 광고)도 옵션으로 도입해 피드에 자연스럽게 녹일 수 있습니다. 모든 광고는 '광고(Advertisement)' 레이블을 명시하여 콘텐츠와 혼동되지 않도록 해야 합니다 [OBJ].
- 정책 준수: 광고는 콘텐츠 영역 근처에 배치하되, 사용자가 콘텐츠 탐색에 방해되지 않도록 배치합니다 [OBJ]. 예를 들어 링크처럼 보이는 광고 텍스트나 이미지를 콘텐츠로 오인하지 않도록 주의해야 합니다.
- 구글 승인을 위한 준비: AdSense 신청 전 독자적이고 고품질의 콘텐츠(고유성)를 확보해야 하며, 만 18세 이상이어야 합니다 [OBJ] [OBJ]. 서비스 소유자(도메인 소유

자)가 사이트의 HTML 소스에 접근할 수 있어야 하므로, 도메인 등록과 <head>에 스크립트 삽입이 가능한 형태로 배포해야 합니다 [🔗](#) [🔗](#). AdSense 코드(<script async src="...adsbygoogle.js?...">)는 생성 후 모든 페이지의 <head> 태그 안에 삽입합니다 [🔗](#).

- 배치 예시: 피드 최상단, 각 게시글 상단·하단, 게시글 본문 중간 (paragraph 사이)에 광고 단위를 넣습니다. 너무 많은 광고는 피해야 하며, 콘텐츠 양과 광고 양의 균형을 유지합니다. 자동 광고(Auto ads) 기능을 함께 사용하여 최적의 위치를 자동으로 테스트할 수도 있습니다.
- 안정성: 광고 스크립트 삽입 시 Content Security Policy(CSP)도 검토하여 스크립트가 정상 로드되도록 설정하며, 크롬 개발자 도구로 광고 노출 상태를 검증합니다.

배포 및 호스팅

- Vercel 배포: Vercel 플랫폼에 프로젝트를 연결하고 Git 연동 CI/CD를 설정합니다. Next.js/React 프로젝트를 Vercel에 배포하면 추가 설정 없이도 전 세계 엣지 네트워크로 확장되고 고성능을 얻을 수 있습니다 [🔗](#). 즉, 무설정(zero-config) 배포로 자동으로 글로벌 CDN에 캐시되며, 가용성과 응답 속도가 향상됩니다.
- Static 및 Serverless: 기본적으로 SSG(정적 사이트 생성)나 ISR을 활용해 게시판 페이지를 정적으로 생성하고, 필요 시 Vercel Serverless Functions(백엔드 API)로 동적 처리(예: 게시글 조회·생성·댓글 API)를 구현합니다. Vercel Functions를 사용하면 서버리스 방식으로 SSR/CSR을 처리할 수 있어 동적 데이터 제공이 가능합니다 [🔗](#).
- 서버리스 데이터베이스: 앞서 언급한 Vercel KV, Supabase, Firebase 등에서 게시글/댓글 데이터를 저장합니다. 예를 들어 Vercel KV(Upstash)와 Supabase를 함께 사용하면 간단한 키-값 저장소와 관계형 DB를 조합할 수 있습니다. 이미지 파일은 Vercel Blob Storage나 Supabase Storage에 저장하고, 파일 URL을 게시글 메타데이터로 관리합니다.
- CI/CD 파이프라인: GitHub/GitLab 연동을 통해 코드 푸시 시마다 자동 빌드/배포가 이루어지도록 설정합니다. 예를 들어 모든 커밋은 미리보기 URL을 생성하고, 메인 브랜치 푸시 시 운영 환경에 배포되도록 합니다. 환경 변수(AdSense 클라이언트 ID, API 키 등)는 Vercel 대시보드에 안전하게 저장합니다.
- 도메인 및 HTTPS: 커스텀 도메인을 연결하고 무료 SSL 인증서로 HTTPS를 적용합니다. Vercel은 자동으로 SSL을 처리하므로 별도 작업 없이 HTTPS 보안을 갖치게 됩니다.

운영 및 안전성

- 욕설/부적절 언어 필터링: 댓글 및 게시글 입력 시 기본적인 욕설 필터를 적용합니다. JavaScript 라이브러리 badwords-ko(한국어 욕설 필터)나 bad-words(영어 욕설 필터) 등을 사용해 금치어를 검출·대체할 수 있습니다 [\[1\]](#). 이 필터는 예를 들어 “욕을 합니다 개새끼”와 같은 문장을 “욕을 합니다 ***”로 변환해 줍니다.
- 자동 차단 및 경고: 욕설이 검출된 댓글은 즉시 차단되거나 관리자 검토 목록에 오르도록 로그를 남깁니다. 초기에는 별도의 관리자 UI 없이도 로그를 파일이나 구글 시트, 슬랙 알림 등으로 처리할 수 있습니다.
- 향후 확장성: 운영 중 비정상적인 사용(스팸 게시글, 지속적 욕설 등)이 늘어나면, 관리자가 직접 모니터링하고 삭제할 수 있는 관리자 페이지나 신고 시스템(신고 버튼)을 추가할 여지를 둡니다.
- 개인정보 보호: 사용자 익명성을 최대한 보장하며, 광고 외에 구글 애널리틱스 같은 최소한의 트래킹만 적용합니다. 개인 식별 정보(PII)는 수집하지 않습니다.

기술 스택 제안

- 프론트엔드 프레임워크: React(또는 Next.js)로 작성하고, Next.js를 사용하면 Vercel 배포 시 이점이 큼니다. React Query/SWR로 게시글·댓글 데이터를 패칭하면 상태 관리가 편리합니다.
- UI 라이브러리: TailwindCSS나 Chakra UI를 이용해 빠른 UI 개발과 반응형 디자인을 구현합니다.
- 폼/입력: 파일 업로드(이미지 첨부)에는 React Dropzone, 이미지 미리보기에는 브라우저 FileReader API를 사용할 수 있습니다.
- 프로비니언시 필터링: 한국어 필터 badwords-ko [\[1\]](#), 영어 필터 bad-words 같은 오픈소스 라이브러리를 적용합니다.
- 서버리스 DB/API: 앞서 언급한 Supabase, Firebase, Vercel KV(Upstash) 등을 사용합니다. 예를 들어 Supabase의 인증 없이 Postgres를 사용하거나, 간단한 카운터는 Vercel KV로 처리할 수 있습니다 [\[2\]](#) [\[3\]](#) [\[4\]](#).
- 광고 통합: Google AdSense 광고 코드는 <script> 형태로 React 컴포넌트 또는 페이지 템플릿의 <head>에 비동기 삽입합니다 [\[5\]](#). 광고 단위 배치 라이브러리

는 특별히 필요 없으며, 일단 HTML 마크업 위치에 코드 블록을 넣는 방식으로 구현합니다.

- CI/CD & 호스팅: Vercel 플랫폼에서 Git 연동을 사용해 배포합니다. 필요 시 GitHub Actions를 추가로 구성하여 코드 린팅(ESLint)이나 테스트(Cypress, Jest)를 자동화할 수 있습니다.
- 기타: 부가적으로 Sentry 같은 에러 모니터링, Google Analytics 등을 도입하면 서비스 안정성과 사용자 분석에 도움이 됩니다.

각 기능 구현 시 위에서 언급한 자료 및 가이드를 참고하여 정책을 준수하고 안정적인 서비스를 유지하도록 합니다.