

MOA 크라우드펀딩 플랫폼

📋 프로젝트 개요

MOA는 크리에이터와 서포터를 연결하는 크라우드펀딩 플랫폼입니다. 프로젝트 후원, 리워드 제공, 실시간 진행 상황 추적 등 크라우드펀딩의 전 과정을 지원합니다.

🔗 기술 스택

Frontend

- **React 19.1.1** - 함수형 컴포넌트 기반 UI 라이브러리
- **TypeScript 5.9.3** - 타입 안정성 보장
- **Vite 7.1.7** - 빠른 개발 서버 및 빌드 도구
- **React Router 7.9.5** - 클라이언트 사이드 라우팅
- **Tailwind CSS 4.1.16** - 유틸리티 기반 CSS 프레임워크
- **Zustand 5.0.8** - 경량 상태 관리 라이브러리
- **Axios 1.13.1** - HTTP 클라이언트

Payment Integration

- **@tosspayments/tosspayments-sdk 2.4.1** - 토스페이먼츠 결제 SDK (결제창 SDK 방식)

Development Tools

- **ESLint** - 코드 품질 관리
- **Prettier** - 코드 포매팅
- **TypeScript ESLint** - TypeScript 린팅

🏗️ 아키텍처

디렉토리 구조

```
src/
├── features/           # 도메인별 기능 모듈
│   ├── admin/         # 관리자 기능
│   │   ├── api/       # 관리자 API 서비스
│   │   └── types.ts    # 관리자 타입 정의
│   ├── auth/          # 인증/인가
│   ├── creator/       # 크리에이터 기능
│   ├── maker/         # 메이커 기능
│   ├── projects/      # 프로젝트 관리
│   │   ├── api/       # 프로젝트 API 서비스
│   │   │   ├── projectService.ts    # 카테고리 기반 조회
│   │   │   └── publicProjectsService.ts # 공개 프로젝트 조회
│   └── components/    # 프로젝트 컴포넌트
```

```

├── ProjectCard.tsx      # 일반 프로젝트 카드
├── TrendingProjectCard.tsx # 지금 뜨는 프로젝트 카드
├── PublicProjectCard.tsx # 공개 프로젝트 카드 (벤티지 지원)
├── types.ts            # 프로젝트 타입 정의
├── supporter/          # 서포터 기능
├── uploads/            # 이미지 업로드
├── pages/               # 페이지 컴포넌트
├──   ├── Auth/          # 로그인/회원가입
├──   ├── Payment/        # 결제 페이지
├──   ├── Projects/       # 프로젝트 목록/상세
├──   ├── Profile/        # 프로필 페이지
├──   ├── Admin/          # 관리자 페이지
├──   ├── ReviewConsole/  # 프로젝트 심사 콘솔
├──   └── Home/           # 홈 페이지
├──   └── components/     # 홈 섹션 컴포넌트들
├── router/             # 라우팅 설정
├── services/           # API 서비스
├──   ├── api.ts         # 실제 API 호출
├──   └── apiMock.ts     # Mock API (개발용)
├── shared/             # 공통 컴포넌트/유틸
├──   ├── components/    # 재사용 컴포넌트
├──   ├── utils/         # 유틸리티 함수
├──   ├── payment.ts     # 토스페이먼츠 결제 유틸
├──   └── categoryMapper.ts # 카테고리 매핑
├── hooks/              # 커스텀 훅
├── styles/             # 전역 스타일

```

아키텍처 패턴

1. **Feature-based 구조**: 도메인별로 기능을 모듈화
2. **관심사 분리**: API, 컴포넌트, 상태 관리 분리
3. **재사용성**: 공통 컴포넌트와 유틸리티 중앙 관리
4. **타입 안정성**: TypeScript로 타입 체크
5. **컴포넌트 분리**: 홈 페이지 섹션을 개별 컴포넌트로 분리

❖ 주요 기능

1. 인증/인가 시스템

구현 기능

- 이메일 회원가입/로그인
- 소셜 로그인 (Google, Kakao)
- JWT 기반 인증
- 이메일 인증
- 역할 기반 접근 제어 (USER, ADMIN)
- 회원가입 후 자동 로그인 제거 (로그인 화면으로 리다이렉트)

주요 컴포넌트

- **LoginPage** - 로그인 페이지
- **SignupPage** - 회원가입 페이지 (자동 로그인 없이 로그인 화면으로 이동)
- **OAuthCallbackPage** - 소셜 로그인 콜백 처리
- **RequireAuth** - 인증 필요 라우트 가드
- **RequireAdmin** - 관리자 권한 라우트 가드

2. 홈 화면 섹션 시스템

홈 페이지 섹션 구성

1. 당신을 위한 추천 (로그인된 사용자에게만 표시)

- 개인화 추천 프로젝트
- 로그인 상태에 따라 조건부 렌더링

2. 🔥 지금 뜨는 프로젝트

- API: **GET /public/projects/trending?size=6**
- 찜(북마크) 많은 순으로 정렬
- **TrendingProjectResponseDTO** 사용
- 진행 상태 배지 표시 (진행중, 공개 예정)

3. ⏰ 곧 마감되는 프로젝트

- API: **GET /public/projects/closing-soon**
- LIVE + APPROVED + endDate 7일 이내 프로젝트
- "마감 임박" 배지 자동 표시

4. 🆕 방금 업로드된 신규 프로젝트

- API: **GET /public/projects/newly-uploaded?size=6**
- 최근 3일 이내 생성된 프로젝트
- "신규" 배지 자동 표시

5. 🔍 지금 많이 보고 있는 프로젝트

- API: **GET /project/category?category=TECH&sort=popular&size=6**
- 뷰 기준 인기 프로젝트

6. 🎯 목표 달성에 가까운 프로젝트

- API: **GET /public/projects/near-goal?size=6**
- 달성률 70~95% 프로젝트
- 달성률 퍼센트 및 진행 바 표시

7. ✅ 성공 메이커의 새 프로젝트

- API: **GET /public/projects/success-maker-new?size=6**
- 과거 성공 이력 있는 메이커의 새 프로젝트
- "성공 메이커" 배지 자동 표시

8. 🏆 첫 도전 메이커 응원하기

- API: `GET /public/projects/first-challenge?size=6`
- 첫 프로젝트를 진행 중인 메이커
- "첫 도전" 배지 자동 표시

섹션 컴포넌트 구조

- 각 섹션은 독립적인 컴포넌트로 분리
- 자체 API 호출 및 로딩/에러 상태 관리
- 빈 리스트 상태 처리

3. 프로젝트 관리

크리에이터 기능

- **프로젝트 생성** (위저드 방식)
 - 기본 정보 입력
 - 목표 금액 설정
 - 스토리 작성
 - 리워드 설정
- **프로젝트 대시보드**
- **리워드 관리**
- **프로젝트 상태 관리** (DRAFT, REVIEW, APPROVED, SCHEDULED, LIVE, ENDED, REJECTED)

서포터 기능

- **프로젝트 탐색** (카테고리, 검색, 정렬)
- **프로젝트 상세 보기**
- **후원하기** (리워드 선택, 배송 정보 입력)
- **후원 내역 조회**

관리자 기능

- **프로젝트 심사 콘솔** (`/admin/review`)
 - 심사 대기 프로젝트 목록
 - 프로젝트 상세 검토
 - 자동 점검 결과 표시
 - 체크리스트 기능
 - 승인/반려 처리
 - 반려 사유 작성
- **프로젝트 목록 관리**
- **프로젝트 승인/거부**

4. 카테고리 시스템

카테고리 타입 및 규칙

- **ProjectCategory 타입**: 백엔드 Category enum과 1:1 대응

```
export type ProjectCategory =  
  | "TECH" | "DESIGN" | "FOOD" | "FASHION"  
  | "BEAUTY" | "HOME_LIVING" | "GAME" | "ART" | "PUBLISH";
```

- **PROJECT_CATEGORY_LABELS**: UI 한글 라벨 매핑

```
export const PROJECT_CATEGORY_LABELS: Record<ProjectCategory, string> = {  
  TECH: "테크.가전",  
  DESIGN: "디자인",  
  FOOD: "푸드",  
  // ...  
};
```

카테고리 처리 규칙

- 백엔드로 전송 시: enum 문자열 그대로 전송 (TECH, DESIGN 등)
- UI 표시 시: **PROJECT_CATEGORY_LABELS** 사용
- `/project/category` API 호출 시 **category** 파라미터 필수

5. 프로젝트 카드 시스템

카드 컴포넌트 종류

1. **ProjectCard**: 일반 프로젝트 카드
 - 모금액, 후원자 수, 진행률 표시
 - 상태 배지 (오른쪽 위)
2. **TrendingProjectCard**: 지금 뜨는 프로젝트 전용
 - 찜 개수 표시
 - 진행 상태 배지 (오른쪽 위)
3. **PublicProjectCard**: 공개 프로젝트 카드 (배지 지원)
 - 배지 시스템 (신규, 마감 임박, 성공 메이커, 첫 도전)
 - 달성률 표시 (achievementRate 있을 때)
 - 모든 배지는 썸네일 오른쪽 위에 배치

배지 시스템

- **신규** (**badgeNew**): 방금 업로드된 프로젝트
- **마감 임박** (**badgeClosingSoon**): 곧 마감되는 프로젝트
- **성공 메이커** (**badgeSuccessMaker**): 성공 메이커의 새 프로젝트
- **첫 도전** (**badgeFirstChallengeMaker**): 첫 도전 메이커 프로젝트

6. 결제 시스템

토스페이먼트 연동

- 결제창 SDK 방식 (위젯 키 불필요)
- 다양한 결제 수단 지원 (카드, 카카오페이, 네이버페이 등)
- 모바일 환경 대응: `windowTarget: "self"` 설정으로 앱 intent 오류 방지
- 결제 승인 프로세스
- 결제 실패 처리

결제 플로우

1. 주문 생성 (POST /api/orders)
2. 토스 결제창 로드 (@tosspayments/tosspayments-sdk)
3. 사용자 결제 수단 선택 및 결제
4. 토스 `successUrl`로 리다이렉트
5. 결제 승인 API 호출 (POST /api/payments/confirm)
6. 백엔드: 토스 승인 API 호출 + Wallet 업데이트

주요 페이지

- `PaymentPage` - 결제 페이지
- `PaymentSuccessPage` - 결제 성공 페이지 (영수증 링크 표시)
- `PaymentFailPage` - 결제 실패 페이지

7. 프로필 관리

서포터 프로필

- 후원 내역 조회
- 프로필 설정
- 공개 프로필 페이지

메이커 프로필

- 프로젝트 목록 (상태별 필터링)
- 프로필 설정
- 공개 프로필 페이지 (프로젝트, 뉴스, 정보)

8. 알림 시스템

- 알림 센터
- 실시간 알림 수신



결제 시스템 상세

기술 구현

결제창 SDK 통합

```
// src/shared/utils/payment.ts
export async function requestTossPayment(opts: {
  orderCode: string;
  amount: number;
  orderName: string;
}) {
  const clientKey = import.meta.env.VITE_TOSS_CLIENT_KEY;
  const customerKey = "user-" + localStorage.getItem("userId");

  const tossPayments = await loadTossPayments(clientKey);
  const payment = tossPayments.payment({ customerKey });

  await payment.requestPayment({
    method: "CARD",
    amount: { currency: "KRW", value: opts.amount },
    orderId: opts.orderCode,
    orderName: opts.orderName,
    successUrl: `${window.location.origin}/payment/success`,
    failUrl: `${window.location.origin}/payment/fail`,
    windowTarget: isMobile ? "self" : "iframe", // 모바일 대응
  });
}
```

API 연동

- 주문 생성: `POST /api/orders`
- 결제 승인: `POST /api/payments/confirm`
 - Request: { `paymentKey`, `orderId`, `amount` }
 - Response: { `paymentId`, `orderId`, `paymentKey`, `method`, `amount`, `status`, `approvedAt`, `receiptUrl` }

보안

- JWT 토큰 기반 인증
- 결제 금액 검증 (프론트엔드와 백엔드 이중 검증)
- 중복 승인 방지

주요 페이지

공개 페이지

- `/` - 홈 페이지 (8개 섹션)
- `/projects` - 프로젝트 목록 (카테고리, 검색, 정렬)
- `/projects/:id` - 프로젝트 상세
- `/projects/:id/pledge` - 후원하기
- `/supporters/:userId` - 서포터 공개 프로필
- `/makers/:makerId` - 메이커 공개 프로필

인증 필요 페이지

- `/profile` - 내 프로필
- `/settings/account` - 계정 설정
- `/settings/supporter` - 서포터 설정
- `/settings/maker` - 메이커 설정
- `/creator/dashboard` - 크리에이터 대시보드
- `/creator/projects/new` - 프로젝트 생성
- `/creator/rewards` - 리워드 관리
- `/notifications` - 알림 센터

관리자 페이지

- `/admin` - 관리자 대시보드
- `/admin/review` - 프로젝트 심사 콘솔 (신규)
- `/admin/projects` - 프로젝트 관리
- `/admin/projects/:projectId` - 프로젝트 상세 검토



인증 및 권한 관리

인증 방식

- **JWT 토큰** (Access Token)
- **Axios** 인터셉터를 통한 자동 토큰 추가
- 토큰 만료 시 자동 로그아웃

권한 레벨

1. **비인증 사용자**: 프로젝트 조회만 가능
2. **USER**: 후원, 프로필 관리 가능
3. **ADMIN**: 프로젝트 승인/거부, 전체 관리 가능

회원가입 플로우

- 회원가입 성공 시 **자동 로그인 없음**
- 로그인 화면으로 리다이렉트
- 사용자가 직접 로그인해야 함



상태 관리

Zustand Store 구조

- `authStore` - 인증 상태 관리
- `projectStore` - 프로젝트 상태 관리
- `makerStore` - 메이커 정보 관리
- `supporterStore` - 서포터 정보 관리
- `myProjectsStore` - 내 프로젝트 목록 관리

API 상태 관리

- **Axios** 인스턴스 중앙 관리
- 요청/응답 인터셉터로 공통 처리
- 에러 핸들링 통합 관리

🧠 UI/UX 특징

디자인 시스템

- **Tailwind CSS** 기반 유틸리티 클래스
- 반응형 디자인 (모바일/데스크톱 지원)
- 일관된 컴포넌트 스타일

주요 컴포넌트

- **Container** - 공통 레이아웃 컨테이너
- **Header** - 네비게이션 헤더
- **Footer** - 푸터
- **ProjectCard** - 프로젝트 카드
- **TrendingProjectCard** - 지금 뜨는 프로젝트 카드
- **PublicProjectCard** - 공개 프로젝트 카드 (배지, 달성률 지원)
- **RewardCard** - 리워드 카드
- **ProgressBar** - 진행률 표시 바

배지 시스템

- 모든 배지는 카드 썸네일 오른쪽 위에 배치
- 배지 종류: 신규, 마감 임박, 성공 메이커, 첫 도전
- 색상 구분으로 시각적 구분

📦 데이터 흐름

홈 페이지 데이터 로딩

1. HomePage 마운트
↓
2. 각 섹션 컴포넌트별 독립적 API 호출
↓
3. 로딩 상태 관리
↓
4. 데이터 표시 또는 빈 상태 메시지

프로젝트 생성 플로우

1. CreatorWizardPage
↓
2. useCreatorWizard Hook (상태 관리)
↓
3. API 호출 (createProject)
↓
4. 백엔드 저장
↓
5. 관리자 승인 대기

후원 플로우

1. ProjectDetailPage → PledgePage
↓
2. 리워드 선택 + 배송 정보 입력
↓
3. 주문 생성 (POST /api/orders)
↓
4. PaymentPage → 토스 결제창
↓
5. 결제 완료 → PaymentSuccessPage
↓
6. 결제 승인 (POST /api/payments/confirm)
↓
7. Wallet 업데이트 (백엔드)

관리자 심사 플로우

1. Admin Review Console 접근
↓
2. 심사 대기 프로젝트 목록 조회 (GET /admin/reviews/projects)
↓
3. 프로젝트 선택 → 상세 정보 조회 (GET /admin/reviews/projects/:id)
↓
4. 자동 점검 결과 확인
↓
5. 체크리스트 작성 (선택)
↓
6. 승인/반려 처리
 - 승인: POST /admin/reviews/projects/:id/approve
 - 반려: POST /admin/reviews/projects/:id/reject

API 엔드포인트

공개 프로젝트 API (인증 불필요)

- `GET /public/projects/trending?size=6` - 지금 뜨는 프로젝트
- `GET /public/projects/closing-soon` - 곧 마감되는 프로젝트
- `GET /public/projects/newly-uploaded?size=6` - 방금 업로드된 신규 프로젝트
- `GET /public/projects/success-maker-new?size=6` - 성공 메이커의 새 프로젝트
- `GET /public/projects/first-challenge?size=6` - 첫 도전 메이커 프로젝트
- `GET /public/projects/near-goal?size=6` - 목표 달성에 가까운 프로젝트

프로젝트 조회 API

- `GET /project/category?category={category}&sort={sort}&page={page}&size={size}` - 카테고리별 프로젝트 조회
 - `category`: 필수 (TECH, DESIGN, FOOD 등)
 - `sort`: popular, progress, latest, closing

관리자 API

- `GET /admin/reviews/projects` - 심사 대기 프로젝트 목록
- `GET /admin/reviews/projects/:id` - 프로젝트 상세 정보
- `POST /admin/reviews/projects/:id/approve` - 프로젝트 승인
- `POST /admin/reviews/projects/:id/reject` - 프로젝트 반려

결제 API

- `POST /api/orders` - 주문 생성
- `POST /api/payments/confirm` - 결제 승인
- `POST /api/payments/:paymentId/cancel` - 결제 취소

배포 및 환경 설정

환경 변수

```
# 백엔드 API 주소
VITE_API_BASE_URL=http://3.35.24.48:8080

# 토스페이먼트 클라이언트 키
VITE_TOSS_CLIENT_KEY=test_ck_xxxxx

# Mock API 사용 여부
VITE_USE MOCK_API=false
```

빌드 및 배포

```
# 개발 서버 실행
npm run dev

# 프로덕션 빌드
npm run build
```

```
# 빌드 미리보기  
npm run preview
```

성능 최적화

구현된 최적화

- **코드 스플리팅**: React Router 기반 라우트별 분리
- **이미지 최적화**: 이미지 업로드 및 표시 최적화
- **API 캐싱**: 검색 결과 캐싱
- **지연 로딩**: 필요 시점에 컴포넌트 로드
- **섹션별 독립 로딩**: 홈 페이지 섹션별 독립적 API 호출

보안 고려사항

구현된 보안 기능

- **JWT 토큰 관리**: localStorage 저장 및 자동 갱신
- **HTTPS 통신**: 프로덕션 환경 HTTPS 사용
- **XSS 방지**: React의 기본 XSS 방지 기능 활용
- **CSRF 방지**: SameSite 쿠키 설정 (백엔드)
- **결제 금액 검증**: 프론트엔드와 백엔드 이중 검증

코드 품질

코드 스타일

- **TypeScript Strict 모드**: 타입 안정성 최대화
- **ESLint**: 코드 품질 관리
- **Prettier**: 일관된 코드 포매팅
- **한글 주석**: 주요 기능에 대한 한글 설명

코딩 컨벤션

- **함수형 컴포넌트**: 클래스 컴포넌트 사용 안 함
- **커스텀 훅**: 로직 재사용
- **타입 정의**: 모든 API 응답/요청 타입 정의
- **에러 처리**: 통합 에러 핸들링
- **컴포넌트 분리**: 큰 컴포넌트를 작은 단위로 분리

주요 성과

구현 완료 기능

- ☑ 사용자 인증/인가 시스템
- ☑ 프로젝트 CRUD 기능
- ☑ 결제 시스템 통합 (토스페이먼츠)
- ☑ 프로필 관리
- ☑ 관리자 기능 (심사 콘솔 포함)
- ☑ 알림 시스템
- ☑ 반응형 UI/UX
- ☑ 홈 화면 섹션 시스템 (8개 섹션)
- ☑ 카테고리 시스템 (타입 안정성 보장)
- ☑ 배지 시스템 (신규, 마감 임박, 성공 메이커, 첫 도전)
- ☑ 달성률 표시 기능
- ☑ 회원가입 플로우 개선 (자동 로그인 제거)
- ☑ 로그인 상태 기반 UI 표시

기술적 성과

- **타입 안정성:** TypeScript로 런타임 에러 최소화
- **모듈화:** Feature-based 구조로 유지보수성 향상
- **재사용성:** 공통 컴포넌트 및 유틸리티 중앙 관리
- **확장성:** 도메인별 모듈 분리로 기능 추가 용이
- **컴포넌트 분리:** 홈 페이지 섹션을 독립 컴포넌트로 분리
- **API 구조화:** 공개 프로젝트 API와 일반 API 분리

🔮 향후 개선 사항

단기 개선

- ☐ 이미지 최적화 (WebP 변환)
- ☐ 무한 스크롤 구현
- ☐ 검색 필터 고도화
- ☐ 에러 바운더리 추가
- ☐ 전체 카테고리 조회 API 추가

장기 개선

- ☐ PWA 지원
- ☐ 오프라인 모드
- ☐ 실시간 알림 (WebSocket)
- ☐ 다국어 지원 (i18n)
- ☐ 개인화 추천 알고리즘 구현

📞 연락처 및 리소스

관련 문서

- [README_PAYMENT.md](#) - 결제 시스템 가이드
- [docs/PAYMENT_FLOW.md](#) - 결제 플로우 상세 설명

- [docs/TOSS_WIDGET_TROUBLESHOOTING.md](#) - 토스 위젯 문제 해결
- [docs/ADMIN_REVIEW_CONSOLE.md](#) - 관리자 심사 콘솔 가이드

기술 문서

- React 공식 문서: <https://react.dev>
 - TypeScript 공식 문서: <https://www.typescriptlang.org>
 - Tailwind CSS 문서: <https://tailwindcss.com>
 - 토스페이먼츠 개발자센터: <https://developers.tosspayments.com>
-

작성일: 2025년 1월

버전: 2.0.0