



프로젝트 개요

Master Board라는 이름의 다기능 게시판 시스템으로, 일반게시판, 갤러리, 관리자게시판으로 구성된 웹 애플리케이션입니다.

개발 필요성 및 배경

- 정보 중앙화 및 효율성 강화

기존 프로젝트 관리는 여러 시스템과 도구에 분산되어 있어 일정 중복, 리소스 충돌, 우선순위 혼란 등이 빈번하게 발생합니다. Master Board는 모든 프로젝트 정보를 한곳에 모아 '단일 정보 소스(Single Source of Truth)'를 제공함으로써 협업과 의사결정을 신속하게 지원합니다.

- 복잡한 일정 관리 및 실시간 추적 지원

다양한 프로젝트의 동시 진행과 일정 변동성을 실시간 대시보드와 맞춤형 도구로 관리하여, 관리자 및 팀원이 현재 상태를 즉각적으로 확인하고 유연하게 대응할 수 있습니다.

- 팀 간 협업과 소통 최적화

업무 분담과 진행 현황, 이슈 공유가 명확해져 소통 장애가 줄고 업무 중복이 최소화되어 팀워크가 향상됩니다.

- 마감 준수 및 생산성 제고

실시간 마감일 및 진행률 확인으로 일정 지연을 방지하며, 리소스 할당 조정으로 전반적인 프로젝트 성공률을 높입니다.

- 자동화 및 데이터 기반 의사결정 지원


반복적인 알림, 승인, 보고서 작성 등이 자동화되어 업무 부담을 줄이고, 실시간 데이터 분석을 통한 빠르고 정확한 의사결정을 가능하게 합니다.

- 검증된 성공 사례 기반 설계

monday.com, Atlassian Jira 등 글로벌 협업 플랫폼이 중앙화, 자동화, 시각화 기능을 통해 팀 생산성과 효율을 극대화한 사례들을 참고하였습니다.

CenterWeb - Django 게시판 시스템

Django 기반의 웹 게시판 시스템 개발 프로젝트



 **목표 개발 일정: 2024년 7월 25일 (과제 마감일)**

팀 구성 및 역할

- **박수연:** 프로젝트 매니저(PM), 개발 리드, 일정관리, 산출물 양식 제공, 템플릿 개발, 모듈 통합, 리빌드 총괄
- **진범광:** 로그인/로그아웃 시스템, 프론트엔드 HTML 개발
- **박태응:** 게시판 시스템

17 개발 일정


7월 14일 (월) - 기획 단계

- 화면 기획서 생성
-  화면 기획서
-  **전체 오픈 전에 문서 권한을 수정 금지로 변경 필요**

7월 15일 (화) - 공통 작업

- **박수연:** 템플릿 가이드 작업
- **진범광:** 로그인/로그아웃 시스템
- **박태응:** 게시판 시스템


7월 17일 (목) - 설계 단계

- **정보구조도 작성** (API, 모듈 메소드 정의)
- **공통 작업:** 박수연(템플릿), 진범광(로그인/로그아웃), 박태응(게시판)
-  정보구조도



7월 18일 (금) ~ 7월 21일 (월) - DB 설계

- **ERD 테이블 설계**
- **참여자:** 박수연, 진범광, 박태응

7월 22일 (화) - 시스템 설계

- Django 기반 개발 협의
- 시퀀스 다이어그램 설계
-  시퀀스 다이어그램
- 공통 작업: 박수연(템플릿), 진범광(로그인/로그아웃), 박태응(게시판)

7월 23일 (수) - 개발 시작

- Django 기반 개발 시작
- 시퀀스 다이어그램 기반 모듈 개발
- 개발자: 진범광(로그인/로그아웃), 박태응(게시판)
- **프론트엔드 HTML 개발:** 진범광 (전체 HTML 템플릿 작업) - <https://github.com/JinBeomKwang/-html->
-  **모듈 테스트 완료**
-  **문제 발생:** 모듈 연계 과정에서 Django ORM과 MariaDB 데이터 USER 베이스 쿼리 충돌 문제
- 디버깅 담당: 박수연, 박태응, 진범광

7월 24일 (목) - 리빌드

- **HTML 템플릿 전달:** 진범광 → 박수연 (목요일까지 작업 완료)
 - **전체 모듈 통합:** 박수연이 모든 모듈을 받아서 통합 작업
 - **ORM 기반 리빌드 결정**
 - **Django 라이브러리 기반으로 회원 모듈 변경**
 - **어드민 APP, BOARD APP 분할**
 - **순차 모듈 결합 작업:** 박수연 (리빌드 총괄)
1. 회원가입/로그인 (기반 시스템)
 2. User 모델 + 프로필 (데이터 모델)
 3. HTML 디자인 통합 (프레젠테이션 레이어)
 4. 보드 앱 생성 (모듈 분리)
 5. 게시판 → 갤러리 (핵심 비즈니스 로직)

- 6. 삭제/업로드 (세부 기능)
- 7. 경로 수정 (라우팅)
- 8. 어드민 페이지 (관리 기능)

- ✅ 1층: 인증 시스템 (Foundation)
- ✅ 2층: 데이터 모델 (Data Layer)
- ✅ 3층: UI 통합 (Presentation Layer)
- ✅ 4층: 비즈니스 로직 (Business Layer)
- ✅ 5층: 관리 기능 (Admin Layer)

1차 개발 완료

7월 25일 (금) - 통합 테스트 및 과제 완료

- 🎯 과제 마감일
- 통합 테스트 QA
- 프리테스트 기반 QA 요청 문서 작성
- QA: 진범광(로그인/로그아웃), 박태응(게시판)
- 최종 제출: 목표 일정 내 성공적 완료 ✅
- 📄 QA 요청 문서

🔧 기술 스택

Backend Framework

- Django 5.2.4 - Python 웹 프레임워크
- Python 3.13.5 - 프로그래밍 언어

Database

- SQLite 3.49.1 - 개발환경 데이터베이스
- Django ORM - 객체 관계 매핑

Frontend

- HTML5 - 마크업 언어
- CSS3 - 스타일시트
- JavaScript (ES6+) - 클라이언트 사이드 스크립팅

- **Responsive Design** - 반응형 웹 디자인

Authentication & Security

- **Django Authentication System** - 내장 인증 시스템
- **CSRF Protection** - Cross-Site Request Forgery 방어
- **User Permission System** - 사용자 권한 관리

Development Environment

- **Windows 11** - 개발 운영체제
- **Git** - 버전 관리 시스템
- **Virtual Environment** - Python 가상환경

Additional Libraries

- **Pillow** - 이미지 처리 라이브러리
- **Django Static Files** - 정적 파일 관리
- **Django Media Files** - 미디어 파일 관리



프로젝트 구조

```
Centerweb/
├── mysite/                # Django 프로젝트 설정
│   ├── __init__.py
│   ├── asgi.py
│   ├── settings.py       # 프로젝트 설정
│   ├── urls.py           # 메인 URL 라우팅
│   └── wsgi.py
├── accounts/             # 사용자 인증 앱
│   ├── migrations/
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── forms.py          # 회원가입/로그인 폼
│   ├── models.py         # 사용자 모델 확장
│   ├── tests.py
│   └── urls.py           # 인증 관련 URL
```

```

|   └─ views.py      # 로그인/로그아웃/회원가입 뷰
├─ board/           # 게시판 앱
|   └─ migrations/
|   └─ __init__.py
|   └─ admin.py
|   └─ apps.py
|   └─ forms.py      # 게시글 작성 폼
|   └─ models.py     # 게시글 모델
|   └─ tests.py
|   └─ urls.py       # 게시판 URL
|   └─ views.py      # 게시판 뷰 로직
├─ media/           # 업로드 파일 저장소
├─ templates/       # HTML 템플릿
|   └─ board/        # 게시판 템플릿
|       └─ admin_board_list.html # 관리자 게시판 목록
|       └─ gallery_form.html    # 갤러리 작성 폼
|       └─ post_confirm_delete.html # 게시글 삭제 확인
|       └─ post_detail.html     # 게시글 상세 보기
|       └─ post_form.html      # 게시글 작성 폼
|   └─ registration/ # 인증 템플릿
|       └─ login.html    # 로그인 페이지
|       └─ profile.html  # 프로필 페이지
|       └─ signup.html   # 회원가입 페이지
|   └─ base.html        # 기본 템플릿
|   └─ board_detail.html # 게시글 상세
|   └─ board_list.html  # 게시판 목록
|   └─ board_write.html # 게시글 작성
|   └─ gallery.html     # 갤러리 목록
|   └─ gallery_detail.html # 갤러리 상세
|   └─ gallery_write.html # 갤러리 작성
|   └─ home.html        # 메인 홈페이지
├─ db.sqlite3          # SQLite 데이터베이스
├─ manage.py           # Django 관리 명령어
└─ README.md           # 프로젝트 문서

```

주요 기능

사용자 인증 시스템

- ☒ **회원가입** - 사용자 등록 및 프로필 생성
- ☒ **로그인/로그아웃** - 세션 기반 인증
- ☒ **프로필 관리** - 개인정보 수정 및 프로필 이미지 업로드
- ☒ **권한 관리** - 일반사용자/관리자 권한 분리
- ☒ **보안 기능** - CSRF 토큰, 비밀번호 해싱

게시판 시스템

- ☒ **일반게시판** - 로그인 사용자 전용 텍스트 게시판
- ☒ **갤러리** - 이미지 업로드 및 미리보기 지원
- ☒ **관리자게시판** - 스태프 권한 필요한 공지사항 게시판
- ☒ **CRUD 기능** - 게시글 생성, 조회, 수정, 삭제
- ☒ **조회수/좋아요** - 게시글 조회수 증가 및 좋아요 기능
- ☒ **페이징 처리** - 대용량 데이터 효율적 표시
- ☒ **검색 및 필터링** - 게시판별 콘텐츠 분류

홈페이지 대시보드

- ☒ **최신글 미리보기** - 각 게시판별 최신 게시글 3개씩 표시
- ☒ **통합 네비게이션** - 일관된 사용자 경험

관리 기능

- ☒ **Django Admin** - 관리자 페이지 통합
- ☒ **사용자 관리** - 회원 정보 및 권한 관리
- ☒ **콘텐츠 관리** - 게시글 및 이미지 관리
- ☒ **데이터베이스 관리** - ORM 기반 데이터 조작

설치 및 실행

시스템 요구사항

- **Python:** 3.13.5 이상
- **Django:** 5.2.4 이상
- **OS:** Windows 11 (개발환경 기준)

설치 가이드

1. 프로젝트 클론

```
git clone [repository-url]  
cd CenterWeb/mysite
```

2. 가상환경 생성 및 활성화

```
python -m venv venv
```

Windows

```
venv\\Scripts\\activate
```

Linux/macOS

```
source venv/bin/activate
```

3. 필수 의존성 설치

```
pip install django==5.2.4  
pip install pillow # 이미지 처리용
```

4. 데이터베이스 설정

```
python manage.py makemigrations accounts  
python manage.py makemigrations board  
python manage.py migrate
```

5. 슈퍼유저 생성 (관리자 계정)

```
python manage.py createsuperuser  
# Username: admin  
# Password: admin123 (예시)
```

6. 개발 서버 실행

```
python manage.py runserver
```

기본 계정 정보

- 관리자 계정: `admin` / `admin123`
- 접속 URL: `http://127.0.0.1:8000/`
- 관리자 페이지: `http://127.0.0.1:8000/admin/`

디렉토리 구조 확인

```
# 프로젝트 구조 확인
cd CenterWeb
tree /F # Windows
# 또는
find . -type f | head -20 # Linux/macOS
```

주요 해결 과제

1. Django ORM과 MariaDB 충돌 문제

- 문제: 사용자 테이블 쿼리 충돌
- 해결: Django 기본 ORM으로 전환, SQLite 사용

2. 모듈 분리 및 통합 과정

- 프로젝트 관리: 박수연의 PM 역할로 일정관리 및 개발 리드
- 표준화: 산출물 양식 제공으로 팀 작업 효율성 향상
- 개발 분담: 각자 담당 모듈 개발
- HTML 템플릿: 진범광이 전체 프론트엔드 담당
- 모듈 통합: 박수연이 모든 모듈을 받아서 통합
- 리빌드 결정: ORM 충돌 문제로 인한 전면 재구성
- accounts 앱: 사용자 인증 전담
- board 앱: 게시판 기능 전담
- 관심사의 분리로 유지보수성 향상

3. 권한 관리 시스템

- 일반 사용자: 일반게시판, 갤러리 작성 가능
- 관리자: 모든 게시판 작성 및 관리 가능



문서 링크

- 화면 기획서
- 정보구조도

- 시퀀스 다이어그램
- QA 요청 문서

프로젝트 성과

- ☒ 목표 일정 내 성공적 완료: 7월 25일 과제 마감일 준수
- ☒ 체계적인 프로젝트 관리: PM 역할로 효율적 일정/프로세스 관리
- ☒ 표준화된 개발 프로세스: 산출물 양식 제공으로 일관된 문서화
- ☒ 시퀀스 다이어그램 기반 체계적 개발
- ☒ Django 표준 아키텍처 적용
- ☒ 모듈별 역할 분담을 통한 효율적 협업
- ☒ 문제 해결 과정에서 기술적 성장
- ☒ 개발 통합 관리로 성공적 프로젝트 완료

개발 기간: 2024년 7월 14일 ~ 7월 25일 (12일)

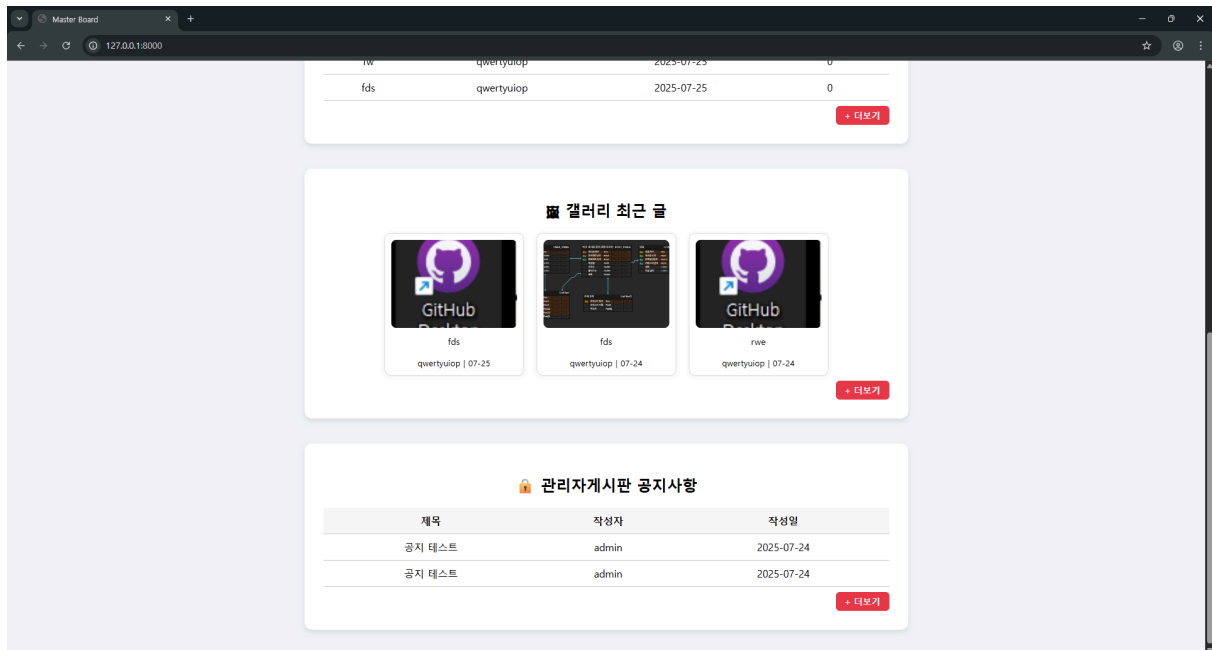
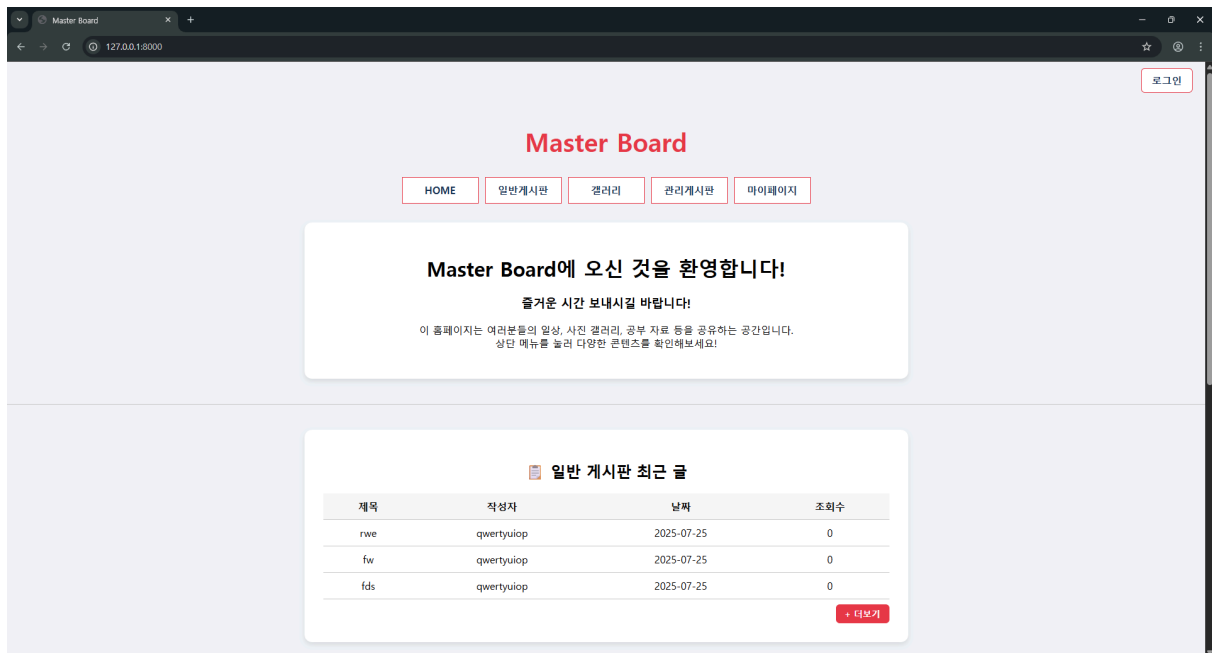
목표 마감일: 2024년 7월 25일 (과제 제출 마감)

개발 인원: 3명

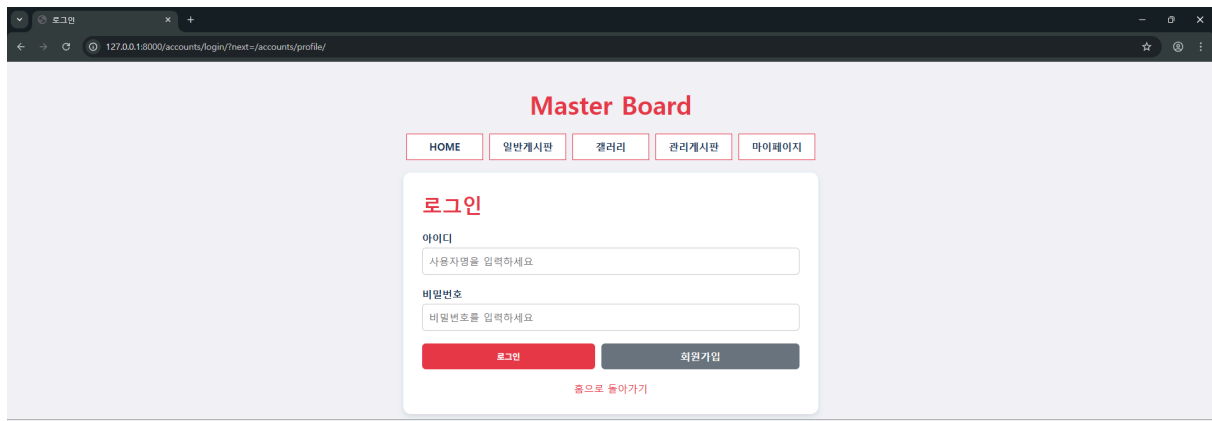
프로젝트 상태: ☒ 목표 일정 내 성공적 완료

시연 사진

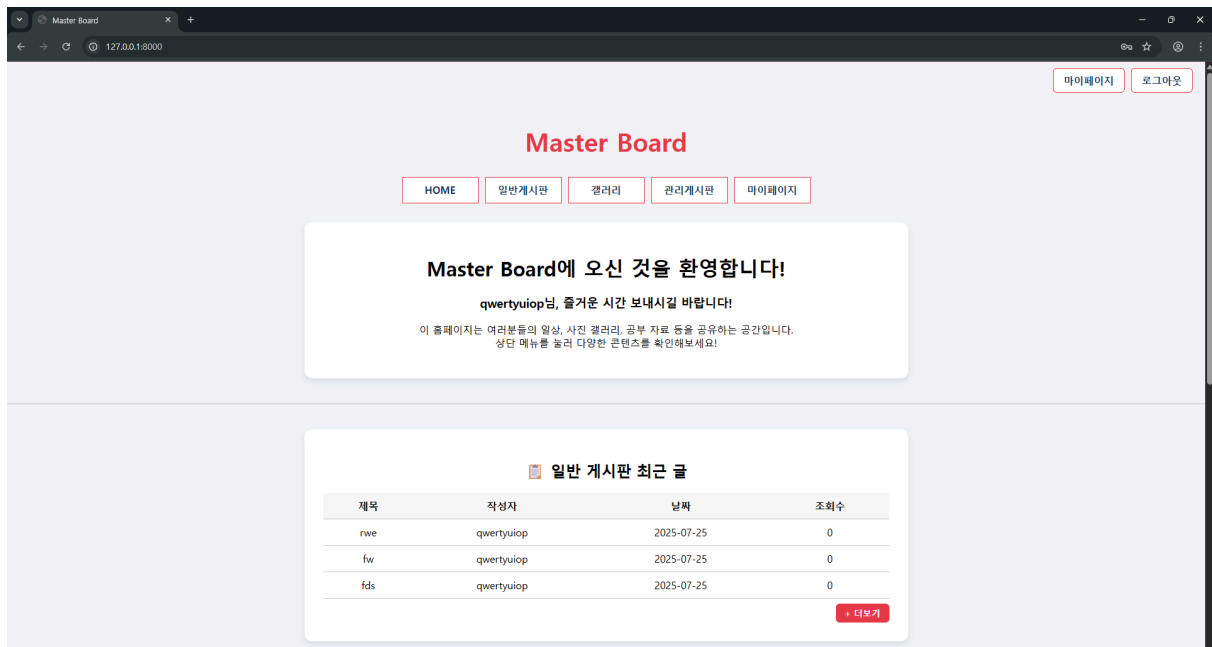
첫화면



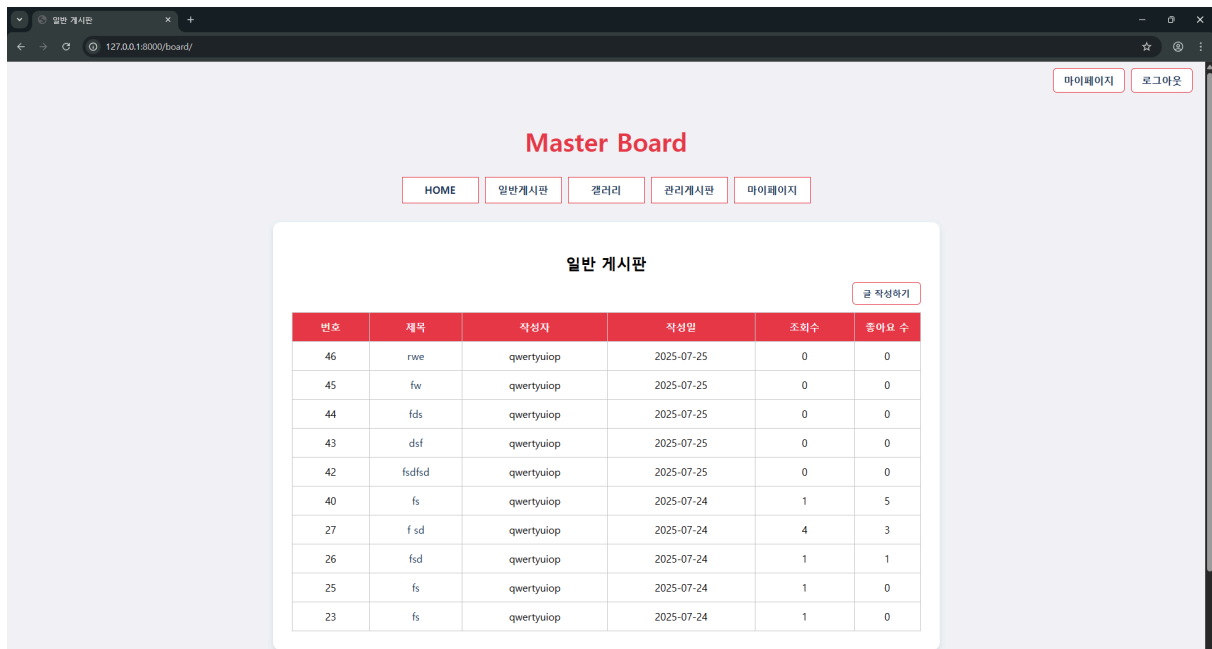
로그인화면



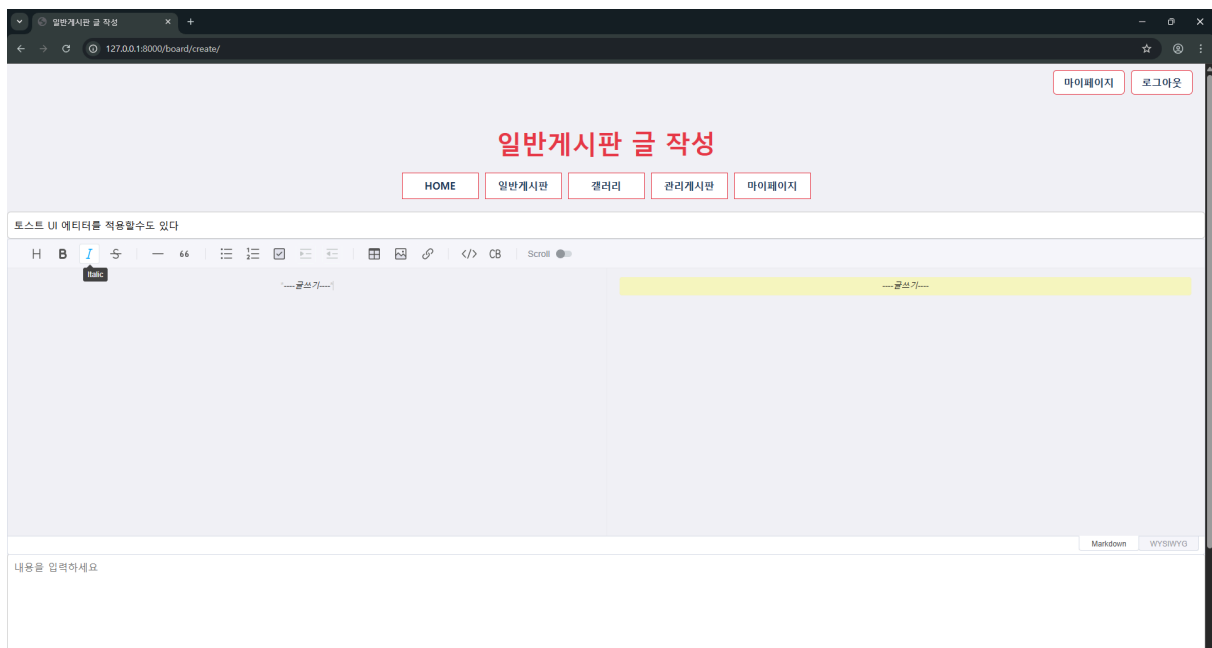
로그인후



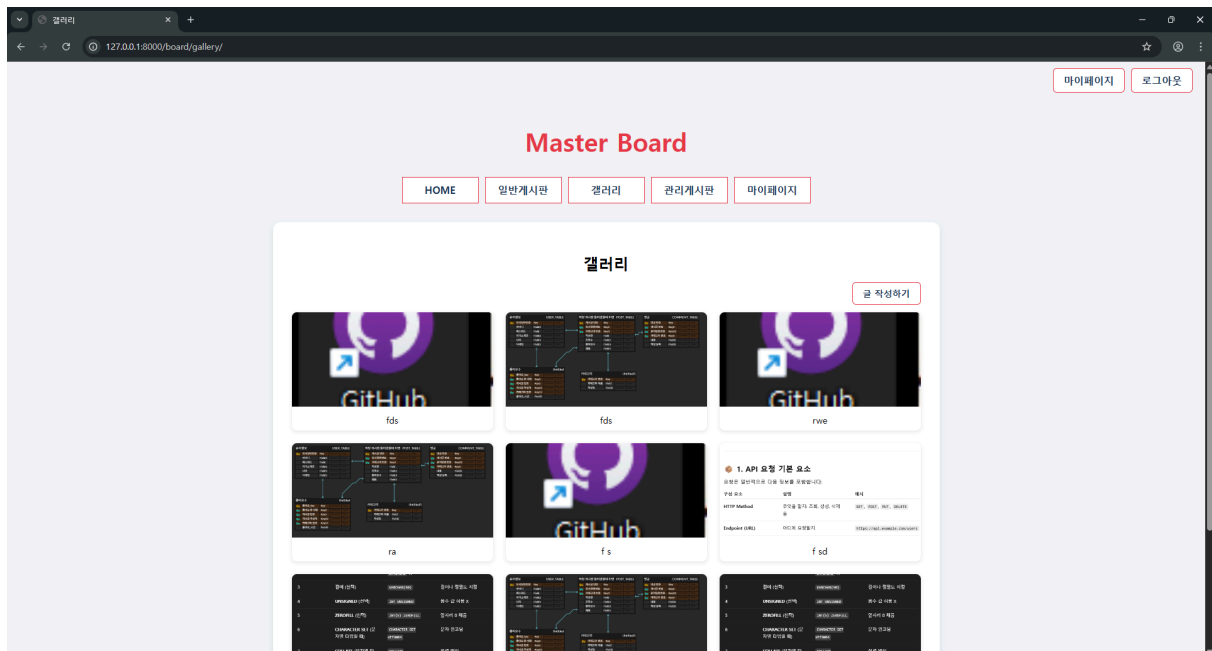
일반게시판



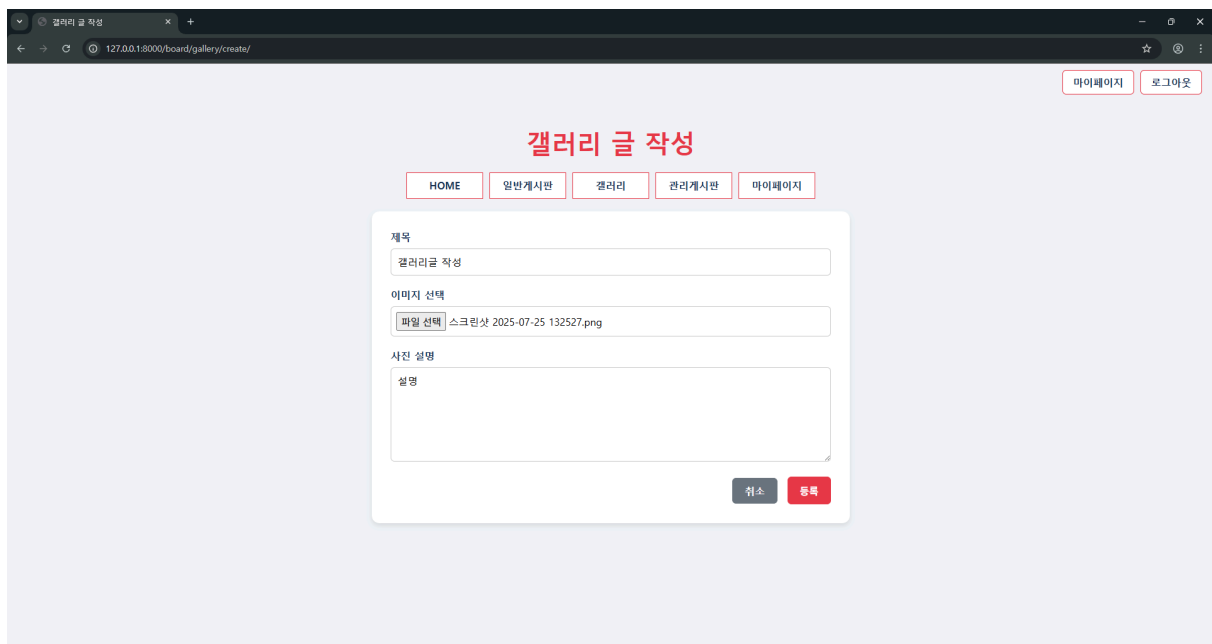
글 작성하기 화면



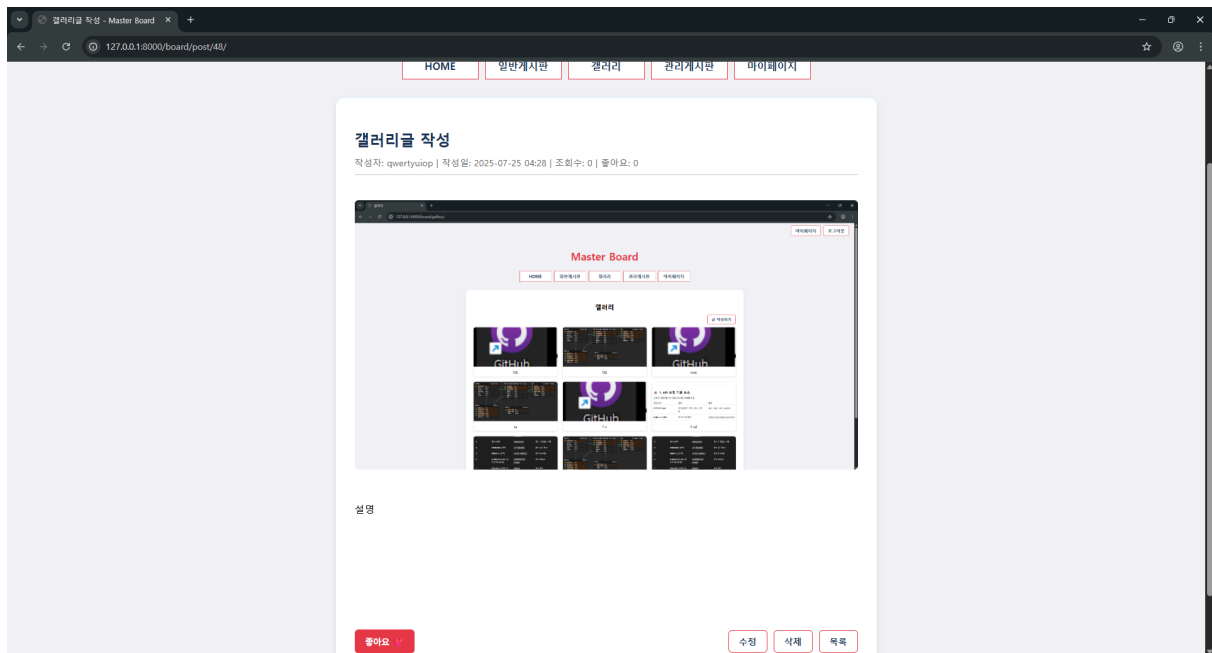
갤러리 화면



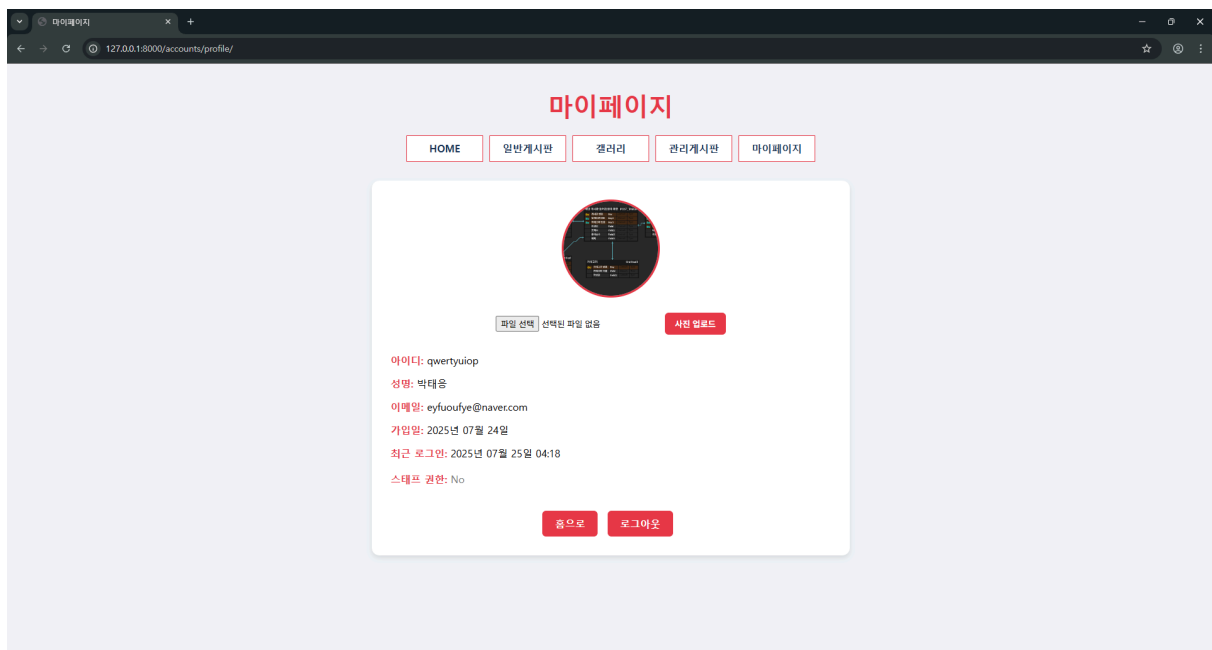
갤러리 글 작성하기 화면



등록하면 상세보기 화면이동



마이페이지 화면



관리자게시판 화면

