광화문마라톤모임 홈페이지 — 현행 시스템 정밀 분석 보고서

문서버전 v1.0 · 작성일 2026-07-05 · 대상 사이트 `http://kwanghwamun.com` 분석 범위: 공개/모바일 화면, 게시판 구조, MySQL DB 전수, 커스텀 봉사점수·페이싱 시스템, 관리자 운영 흐름

---

0. 요약 (Executive Snapshot)

항목	현황
플랫폼	**테크노트6(Technote6)** 기반 PHP 게시판 CMS (2011년 전후 구축, 프레임셋 구조)
서버	cafe24 공유호스팅 / nginx + **PHP 5.x** / **MySQL 5.1.45** (2010년 버전, EOL)
DB 규모	**201개 테이블**, 총 데이터 약 **550MB+**, 회원 게시글 25,000+ / 댓글 140,000+
회원	로그인 회원 **771명** (`a_tn3_memberboard_list`)
핵심 자산	① 자유게시판(글25,538/댓글101,373) ② 페이싱신청(글6,137/댓글30,799) ③ 봉사점수(활동1,970/부여39,357)
운영 방식	**관리자·총무·지역팀장 전원 수작업** (댓글 집계, 명단 텍스트 입력, 엑셀 대조)
현대화 흔적	이전 개발자가 **정원·결제·취소 기반 신청 프로토타입**(`pacing_*` 4개 테이블)을 구축하다 미완
근본 문제	① 회원·페이싱회원·봉사점수 **3중 신원체계 분리** ② 신청을 **댓글로 접수** ③ 권한 **하드코딩** ④ EOL 스택 보안·유지보수 리스크

핵심 결론: **데이터(회원·게시판·신청·점수)는 온전히 살아 있고 마이그레이션 가능**하다. 다만 운영의 병목은 기술이 아니라 **"수작업 프로세스"**에 있으며, 이는 신규 시스템에서 자동화로 100% 제거 가능하다. 이미 존재하는 `pacing_*` 프로토타입이 신규 "정원제 신청" 모델의 검증된 설계 근거가 된다.

---

1. 분석 방법 및 접근 경로

경로	수단	확보 내용
공개 웹	HTTP(curl) — 홈/메인프레임/모바일	프레임셋 구조, 메뉴 라벨, 게시판 네이밍
소스코드	**FTP**(kwanghwamun24.cafe24.com)	`lib.php` , 게시판 엔진, **봉사점수 관리자(`/sub_manage`), 페이싱·랭킹 계산(`/technote/other_insert`)**
DB 스키마·데이터	서버측 임시 읽기전용 스크립트(분석 후 삭제 완료)	201개 테이블 전수 스키마·행수·용량, 핵심 테이블 실데이터 표본

> 참고: 종합관리자(`/technote/admin.php`)와 Adminer(`/adminer.php`)는 cafe24 웹방화벽(WAF)이 `server=` · `username=` 쿼리스트링을 차단(406/403)하여 브라우저 자동화가 제한되었다. 이에 DB는 서버측 스크립트로 직접 조회했고, 분석 종료 후 스크립트는 서버에서 삭제(검증: HTTP 404)했다. 실제 운영자는 관리자 화면 자체는 정상 사용 가능하다.

---

2. 기술 스택 및 인프라 현황

2.1 스택

- **CMS**: Technote6 (한국형 PHP 게시판, 나우콤 계열). 각 게시판이 `_list` / `_re` / `_cnt` / `_ad` 4개 테이블로 구성되는 전형적 구조.
- **언어/DB**: PHP 5.x (구형 `mysql_*` 함수 사용 — PHP7+에서 제거된 API), MySQL **5.1.45**(2010년 릴리스, 공식 지원 종료).
- **화면 구조**: 최상위가 `<frameset>` (프레임 기반). 모바일은 UA 감지로 `/technote/kmob.php` 분기.
- **인증**: PHP 세션 (`session_save_path=/home/hosting_users/kwanghwamun24/www/technote/data/session`), 쿠키도메인 `.kwanghwamun.com` .
- **DB 접속정보 저장**: `technote/data/install/setup_mysql.php` 에 `base64(strrev())` 로 난독화 저장(암호화 아님).

2.2 리스크(기술 부채)

1. **EOL 스택**: PHP5·MySQL5.1은 보안패치 종료. 알려진 취약점 노출, TLS/암호화 현대 표준 미대응.
2. **프레임셋 UI**: 검색엔진/모바일/접근성에 부정적. 반응형 아님(뷰포트 고정 `user-scalable=no`).
3. **비밀번호 관리**: 회원 비번 `m_pass varchar(50)` — 해시 강도·방식 확인 필요(레거시 테크노트는 약한 해시 사용 이력). DB접속정보도 난독화 수준.
4. **WAF 회피 불가/관리도구 제약**: 표준 DB 관리도구(Adminer)가 WAF에 막혀, 운영자가 SQL 레벨 대응이 어려움.
5. **백업/버전관리 부재 추정**: 서버에 `.swp` , `copy_` , `_back` , `2017` 등 수기 백업 흔적 다수 → 체계적 백업/릴리스 파이프라인 없음.

---

3. 사이트 화면·메뉴 전수 인벤토리

프레임셋 좌측 메뉴 + 본문 링크에서 추출한 **실제 운영 메뉴**를 게시판/기능 단위로 정리한다. (행수 = DB 실측)

3.1 커뮤니티/소통

메뉴(라벨)	게시판 id	글	댓글	비고
자유 게시판	`kmfreeboard1`	25,538	**101,373**	사실상 커뮤니티의 심장. 데이터·트래픽 최다
(구)자유/운영계획	`kmfreeboard`	1,165	5,361	연도별 운영계획 등 앵커 글
공지 사항	`kmnotice`	806	1,243	페메요청·적립금 등 운영공지
애경사 게시판	`kmnotes`	507	9,306	회원 경조사 — 댓글 활발
후기/포스트	`kmpostscript`	349	1,466	대회 후기
광화문 뉴스	`kmnews`	44	292	
공동구매 게시판	`kmpurchase`	20	96	
스터디	`studyboard`	266	89	
이벤트	`eventboard`	21	181	

3.2 대회/페이싱 (핵심 업무 영역)

메뉴	게시판 id	글	댓글	비고
페이싱 신청	`kmpacingapply`	6,137	**30,799**	★ 대회 페메 신청을 **댓글로 접수**. 웹팀장 수작업 집계의 핵심
페이싱 기획/계획	`kmpacingplan`	2,937	271	대회 기획서·시간대 계획
연도별 페이싱 대회 목록	`kmpacingdb`	18	1	대회 DB 인덱스
페이싱 일정	`kmpacingschedule`	222	14	
페이싱팀 소개	`kmpacingmember` (board)	489	297	팀원 소개 (명단과 별개)
대회 결과/기록	`kmresult`	1,163	2,898	기록 아카이브
대회 홍보	`kmmarathoninfo`	2,024	13	외부 대회 정보

3.3 봉사/하트/재정

메뉴	게시판 id	글	댓글	비고
자원봉사 신청	`kmserviceapply`	477	1,059	★ 자원봉사도 **댓글 접수**
하트마라톤/달려라하니	`kmheart`	796	~	사회공헌(하트) 활동
하트마라톤 신청	`kmheartapply`	70	962	★ 신청 **댓글 접수**
달려라 하니(꿀단지)	`kmhoney`	719	1,946	
재정 현황	`kmfund`	315	210	회비·적립금 공개
운영진 건의사항	`kmproposal`	18	32	

3.4 회원/자료/소개

메뉴	게시판 id	데이터	비고
일반회원 소개	`kmmembers`	121	회원 프로필
회원 등록정보	`memberboard` / `kkkdata`	771	**★ 로그인 회원 마스터**
광화문 자료실	`kmdata`	186	
갤러리(대회/이벤트 사진)	`kmgallery`	547	+ 메인롤링 `kmgalleryrolling` 229
메디컬	`kmmedical`	23	
런에티켓	`kmruntiquette`	58	
마라톤학교 Q&A	`marathonschool`	112	
모임소개/연혁/규약/운영진	`kmbody` , `kmhistory`	고정페이지	프레임 본문형
캘린더	`kkkcalendar`	967	행사 일정

3.5 법인용 홈피(별도 세트, 미완/템플릿)

`kkkbodyonly` , `kkkqna` , `kkkorder` , `kkknotice` , `kkkmedia` 계열에 **"회사설립 개요·대표자 인사말·조직도·이용약관·서비스 소개·찾아오시는 길"** 등 **법인 사이트용 페이지 골격**이 존재한다. → 미팅 안건 "법인용 홈피(내외부 각각)"의 기존 시도 흔적. 현재는 템플릿 수준.

3.6 모바일

`kmob.php` (모바일 홈), `mobhome` / `mobnotice` / `mobqna` 전용 게시판. 데스크톱과 **별도 관리 → 이중 유지보수 부담**.

> 관찰: 메뉴 라벨에 `kkk*` (구 상용 테크노트 데모 잔재)와 `km*` (광화문 커스텀)이 혼재. 2011년 표기("2011년도 운영계획", "2011 페이싱적립금")와 2026년 표기가 공존 → **10년+ 누적된 정보 구조의 지층**. 신규 IA(정보구조) 재설계 필요.

---

4. 데이터베이스 구조 정밀 분석

4.1 테이블 네이밍 체계

- 접두어 `a_tn` + 세대번호 + 게시판명 + 역할.
  - `a_tn1_root_*` : 전역 설정/카운터 (`root_ad` =사이트설정, `root_cnt` =통합카운터)
  - `a_tn2_<board>_*` : 게시판 데이터
  - `a_tn3_memberboard_*` : 회원/쪽지/스크랩
  - `a_tn4_shop*` : 쇼핑(주문서 `kkkorder` 등)
- 게시판 1개 = 4테이블 세트:
  - `_list` — 글 본문/메타(58컬럼: 제목·작성자·id·wdate(unixtime)·hit·secret·첨부 등)
  - `_re` — 댓글(20컬럼: parent·name·id·tbody·wdate)
  - `_cnt` — **접속통계**(74컬럼: 브라우저별/시간대별 카운터 — MSIE_5, Firefox, hou0~hou23 등). 실데이터 가치는 낮으나 용량은 큼
  - `_ad` — 게시판 개별 설정

> 시사점: `_cnt` (통계) 테이블들이 **전체 용량의 상당 부분**(예: `kmfreeboard1_cnt` 64MB, `kkkmain_cnt` 30MB)을 차지. 마이그레이션 시 통계 원본은 **집계 후 폐기 또는 요약 이관** 대상.

4.2 회원 마스터 — `a_tn3_memberboard_list` (771행, 74컬럼)

핵심 컬럼: `no` (PK), `m_id` (로그인ID), `m_name` (실명), `m_nick` (닉네임), `m_pass` , `m_level` (권한레벨 int), `m_group` , `point1/point2/point3` (bigint, 활동/글쓰기 포인트), `tel1/tel2` , `m_mail` , `jumin` (주민 — ⚠️ 개인정보), `birth` , `sex` , `juso` (주소), `last_login` , `login_cnt` , `add1` ~`add10`(사용자정의 필드 10개).

관찰:

- **주민번호(`jumin`) 컬럼 보유** → 개인정보보호법 대응 필수(신규 이관 시 원칙적 폐기/비수집 권장).
- 포인트 필드(point1~3)는 **테크노트 자체 포인트**(글쓰기 적립 등)로, **봉사점수와는 별개 체계**(아래 4.5).
- `add1~add10` 범용필드에 커스텀 정보가 흩어져 있을 가능성 → 이관 전 필드 의미 매핑 필요.

4.3 대회 페이싱 신청(현행) — `kmpacingapply`

- `_list` (6,137): 대회별 "페메 모집" 공지 글.

- `_re` (**30,799**): **신청이 여기 댓글로 쌓임.** 회원이 "하프 4:30 신청합니다", "10km 레페 신청" 식으로 댓글 작성 → 웹팀장이 **눈으로 읽어 시간대별 인원 집계**.
- 구조적 한계: 정원/마감/중복/취소를 **시스템이 모름**. 초과신청·중복·취소가 자연어 댓글에 섞여 **집계 오류·분쟁** 발생. (미팅 안건 "페이싱 정리 자동화 / 웹팀장 업무 자동화"의 직접 원인)

4.4 자원봉사·하트 신청 — `kmserviceapply` (477/1,059), `kmheartapply` (70/962)

동일 패턴. **신청 = 댓글**. 봉사 신청 내역이 게시판 댓글로만 존재 → 집계·명단화가 전부 수기.

4.5 봉사·활동 점수 엔진 (커스텀) — 가장 중요한 자산

테크노트 밖에서 별도 구축된 3개 테이블:

① `km_common` (11행) — 코드 마스터

- `c001` 점수구분: **행사 / 페이싱 / 하트교실 / 기타**
- `l001` 지역구분: **공통 / 수도권 / 충청권 / 영남권 / 호남권 / 제주권**

② `km_action_main` (1,970행) — 활동(이벤트) 원장

컬럼	의미	예시
`main_id`	활동 PK	1994
`category`	점수구분	페이싱
`local`	지역	수도권
`title`	대회/활동명	2026 화성효마라톤대회
`act_date`	활동일	2026-05-05
`point`	**1인당 부여 점수**	페이싱 3, 자원봉사 8
`comment`	**참가자 명단(공백구분 텍스트!)**	"최준복 김광섭 이사랑 …(50명)"
`reg_id`	입력자	opyheart

③ `km_action_detail` (39,357행) — 회원별 점수 부여 `main_id` + `member_no` + `member_id` + `point` (개별 조정, 대개 0) + `reg_date`. 즉 활동 1건에 참가자 N명을 전개한 **부여 원장**.

점수 산정 알고리즘 (원본 `/sub_manage/view_rank.php` 실 SQL):

```
SELECT u.m_name, u.m_group,
  SUM(CASE m.category WHEN '페이싱'   THEN d.point+m.point ELSE 0 END) AS 페이싱점수,
  SUM(CASE m.category WHEN '하트교실' THEN d.point+m.point ELSE 0 END) AS 하트교실점수,
  SUM(CASE WHEN m.category NOT IN('페이싱','하트교실')
                                  THEN d.point+m.point ELSE 0 END) AS 자원봉사점수,
  SUM(d.point+m.point) AS 총점
FROM km_pacingmember u
LEFT JOIN km_action_detail d ON u.no = d.member_no   -- ※버전에 따라 u.m_id = d.member_id
LEFT JOIN km_action_main  m ON m.main_id = d.main_id
                           AND m.act_date LIKE '{연도}%'
GROUP BY u.m_group, u.no
ORDER BY 총점 DESC;
```

- **점수 = Σ(활동 point + 개별조정 point)**, 카테고리 3분류(페이싱/하트교실/자원봉사)로 나눠 집계, **연도별**(act_date LIKE 'YYYY%') 랭킹.
- 산정 대상 회원 = `km_pacingmember`(페이싱 회원 명단), 기수(`m_group`)·연도(`m_year`) 기준.

치명적 부채 — 신원 매핑의 3중 난립: 원본 소스에 동일 랭킹 쿼리가 **3가지 조인 버전**(`u.m_id=d.member_id` / `u.no=d.member_no` / 서브쿼리)으로 공존한다. 이는 **회원 식별자가 테이블마다 달라**(memberboard.m_id=`km803` 문자열 vs km_action_detail.member_id=`124` 숫자 vs member_no) 개발자가 반복적으로 땜빵한 흔적이다. → 점수 누락/오귀속의 상시 위험.

수작업 병목: 관리자(`opyheart` 등)가 활동 등록 시 **참가자 이름을 공백구분 텍스트로 직접 타이핑**(`km_action_main.comment`) → 시스템이 이름을 회원으로 매칭 → `km_action_detail` 전개. **동명이인·오타·미등록자**는 매칭 실패. 39,357건이 이 방식으로 축적.

4.6 페이싱 회원 명단 (또 다른 신원체계)

- `km_pacingmember`(496): `m_group`(기수, 예 24), `m_id`(km2491), `m_num`, `m_local`(지역권), `m_name`, `m_year`(2026), `sex`, `is_valid`. → **점수 산정의 기준 명단**.
- `pacing_member`(221, 레거시): `temp1~temp30` 등 정크 필드 다수 — 구버전, 사실상 사장.
- `km_pacingmember2017`, `copy_km_pacingmember` : 연도 스냅샷/수기 백업.

→ **회원(771) ≠ 페이싱회원(496) ≠ 점수대상**. 세 명단이 **키로 연결되지 않고 이름/수기로 대조**됨. (미팅 안건 6 "봉사점수·페이싱팀소개·회원리스트 통합"의 근거)

4.7 신형 신청 프로토타입 (이미 구축된 현대화 시드) ★

이전 개발자가 **정원·결제·취소 개념의 신청 시스템**을 별도 구축(테크노트와 무관, 테스트 데이터 수준):

테이블	행	핵심 컬럼	의미
`pacing_board`	9	wr_subject, wr_marathon_name/date/place, wr_teamjang, wr_region_team, **wr_status(모집중)**	대회(모집 공고)
`pacing_options`	177	course_type(full_rp/half_rp), pace_time(4:30), pace_time_min, **num_people(정원)**, is_active	**시간대별 정원 슬롯**
`pacing_apply`	65	option_id, member_id, num_people, **payment_status enum(pending/paid/not_paid)**, **cancel_status(active/cancel_requested/canceled)**, role(manager/team_leader)	신청내역
`pacing_comment`	4	wr_board_id, wr_comment	공고 댓글

의의: 사용자가 원하는 **"정원·잔여·마감·결제·취소"** 모델이 이미 **데이터 구조로 검증**되어 있다. 신규 시스템은 이를 **대회(200명 정원) 단위로 일반화**하면 된다. 즉 신규 신청 모듈은 "무에서 창조"가 아니라 **"검증된 프로토타입의 완성·확장"**이다.

4.8 데이터 규모 톱 테이블(마이그레이션 비용 산정 기준)

테이블	행수	용량	이관 우선순위
kmfreeboard1_list	25,538	97MB	★★★ 콘텐츠
kmfreeboard1_re	101,373	38MB	★★★ 댓글
kmpacingapply_re	30,799	18MB	★★★ 신청이력(구조화 이관)
km_action_detail	39,357	1.2MB	★★★ 점수원장
memberboard_list	771	0.3MB	★★★ 회원
*_cnt(통계) 전체	—	250MB+	★ 요약/폐기

---

5. 사용자 경험(UX) 현황 진단

5.1 일반 회원 관점

- **신청이 직관적이지 않음**: "페메 신청"을 하려면 해당 대회 글을 찾아 → 댓글로 자연어 작성. **내가 원하는 시간대에 자리가 있는지 알 수 없음**(공고 본문에 "시간대별 초과 시 레페로 변경될 수 있음" 안내가 반복 등장 = 정원 불투명의 증거).
- **마감/잔여 정보 없음**: 몇 명 남았는지, 언제 마감인지 화면에 없음. 선착순인데 **선착 여부를 신청자가 확인 불가**.
- **내 신청/취소 관리 없음**: 취소하려면 다시 댓글. 내 신청 현황 대시보드 부재.
- **모바일 경험 분리·구식**: 별도 모바일 페이지, 반응형 아님.
- **내 점수/등수 확인 번거로움**: 랭킹 페이지는 있으나 회원·페이싱회원·점수 명단이 분리되어 "내 정보"가 한 곳에 없음.

5.2 운영진(관리자/총무/지역팀장) 관점 — 수작업 지도

업무	현행 수작업	근거
페메 신청 접수	댓글 30,799건을 눈으로 읽어 시간대별 인원 집계	`kmpacingapply_re`
페이싱 정리	시간대·코스별 인원 수기 배정, 초과분 레페 전환	공고 본문 안내문
봉사/활동 점수	**참가자 이름을 공백구분 텍스트로 직접 타이핑** 후 전개	`km_action_main.comment`, `action.php`
점수 집계/랭킹	연도·카테고리별 SQL 실행, 명단 불일치 수기 보정	`view_rank.php` 3중 조인
페메 횟수 관리	회원별 페이싱 참여 횟수 별도 관리	미팅 "페메횟수 자동화"
명단 통합	회원·페이싱회원·점수명단 **이름으로 대조**	3중 신원체계
권한	특정인만 가능(**ID 하드코딩**)	`inc_vol_auth.php` : sjmodels/opyheart/km1102/Admin, m_level≥50

5.3 권한/보안 관점

- 점수 입력 권한이 **소스코드에 회원ID로 하드코딩**(`inc_vol_auth.php`). 인사이동·팀장교체 시 **개발자가 코드를 고쳐야** 함. 지역팀장 위임·감사로그 없음.
- 감사추적: `km_action_*` 에 `reg_id` / `edit_id` 는 있으나 화면상 이력관리 도구 부재.

---

6. 근본 원인 요약 (신규 시스템 설계의 출발점)

#	근본 문제	파생 증상	신규 시스템 해결 방향
P1	**신청을 댓글로 접수**	정원·잔여·마감·중복·취소를 시스템이 모름 → 전량 수기 집계	정원제 신청 엔티티(`pacing_*` 일반화)
P2	**3중 신원체계 분리** (회원/페이싱회원/점수대상)	이름 대조, 점수 오귀속, 통합뷰 부재	단일 회원 마스터 + 역할/기수 속성
P3	**점수 명단 텍스트 입력**	동명이인·오타·매칭실패, 39,357건 수기	신청/참가 데이터에서 **점수 자동 산정**
P4	**권한 하드코딩**	팀장 위임·감사 불가, 코드수정 의존	역할기반 권한(RBAC) + 위임
P5	**EOL 기술스택**(PHP5/MySQL5.1/프레임셋)	보안·모바일·유지보수 리스크	현대 스택 + 반응형 + 백업/릴리스
P6	**데이터·통계 혼재, 백업 임시**	용량 낭비, 정합성 위험	정규화 스키마 + 통계 분리 + 자동 백업

> 다음 문서 **`02_개선_리포트.md`** 에서 위 6개 근본 문제를 미팅 안건(7/2)과 매핑하여, 신규 아키텍처·데이터 마이그레이션·자동화 설계·로드맵을 제시한다.

부록 A. 게시판 전수 목록(요약)

커뮤니티: kmfreeboard1, kmfreeboard, kmnotice, kmnotes, kmpostscript, kmnews, kmpurchase, studyboard, eventboard, kkkcalendar 대회/페이싱: kmpacingapply, kmpacingplan, kmpacingdb, kmpacingschedule, kmpacingmember(board), kmresult, kmmarathoninfo 봉사/하트/재정: kmserviceapply, kmheart, kmheartapply, kmhoney, kmfund, kmproposal 회원/자료/소개: memberboard, kmmembers, kmdata, kmgallery, kmgalleryrolling, kmmedical, kmruntiquette, marathonschool, kmbody, kmhistory, kmmanager, kmcoordi, kmpoll 법인용: kkkbodyonly, kkkqna, kkkorder, kkknotice, kkkmedia 모바일: mobhome, mobnotice, mobqna 커스텀: km_action_main/detail, km_common, km_pacingmember(+2017/copy), pacing_member, pacing_board/options/apply/comment

부록 B. 핵심 식별자 매핑(마이그레이션 유의)

- 회원: `memberboard.no` (숫자PK) ↔ `memberboard.m_id` (예: `km803`) ↔ `km_pacingmember.m_id` (예: `km2491`) ↔ `km_action_detail.member_no` / `member_id`
- **주의**: `member_id` 가 테이블별로 문자열/숫자 혼용. 이관 시 **회원 마스터 단일키로 정규화**하고 매핑 테이블 작성 필수.