

# 프로젝트 결과 요약 보고서

## 1. 프로젝트 명: KT 프랜차이즈TV

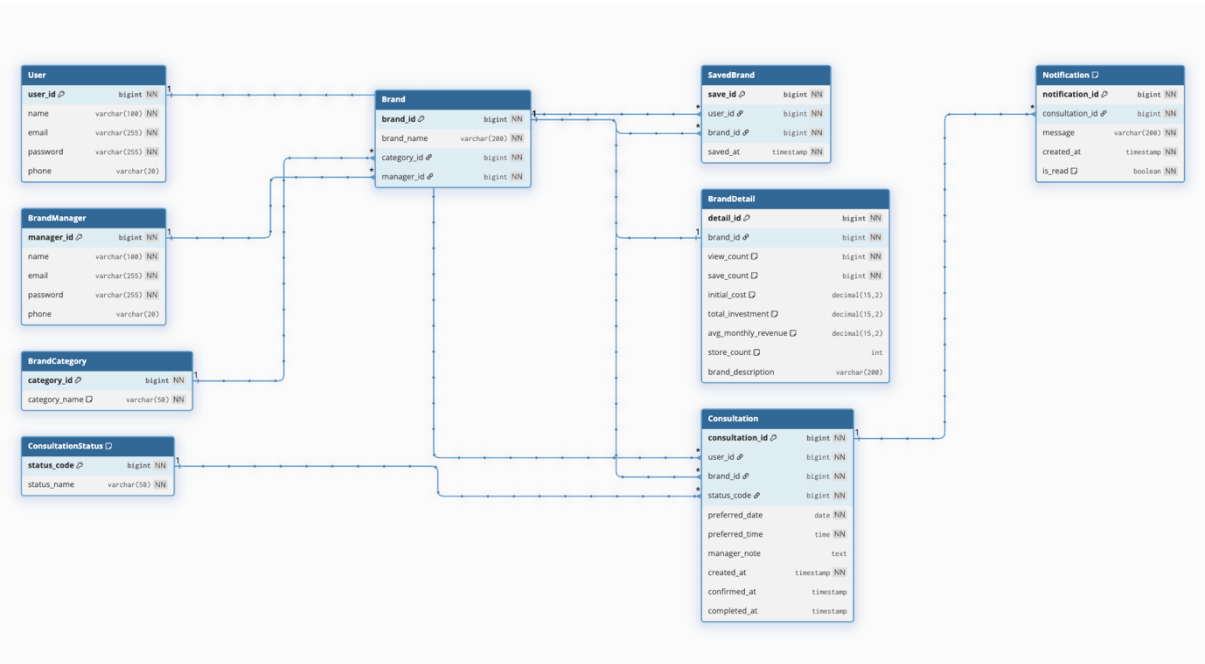
## 2. 프로젝트 개요:

본 서비스는 IPTV 라는, 예비 창업자 포함 일반 사용자들의 접근성이 편리한 채널을 통해 예비 창업자(User)와 프랜차이즈 브랜드(Brand), 브랜드 담당자(BrandManager) 간의 정보 제공과 연결이 손쉽게 일어나는 환경을 조성한다.

비전	누구에게나 손쉬운 프랜차이즈 창업 준비 생태계를 위하여
미션	<p>IPTV 플랫폼을 통해 예비창업자와 가맹업주를 연결하는 원스톱 창업 중개 생태계를 구축하는 것</p> <ul style="list-style-type: none"><li>- 누구나 프랜차이즈 가맹점 창업 정보에 즉시 접근 가능하도록 제공</li><li>- 원클릭으로 전문 창업 컨설팅 상담을 연결</li><li>- 데이터 기반의 창업 의사결정을 지원</li></ul>
핵심가치	<p><b>1) 접근성</b> : 누구나 직관적으로 쉽게 창업 정보에 접근할 수 있는 오픈 플랫폼</p> <p><b>2) 신뢰성</b> : 가맹비, 창업 비용, 평균 매출액, 매장 수 등의 데이터를 명확히 공개하여 신뢰도 높은 정보를 얻을 수 있는 환경</p> <p><b>3) 편의성</b> : 정보 탐색부터 창업 상담 예약까지 바로 가능한 원스톱 서비스</p>
주요 기능	1) 회원가입/로그인/로그아웃
	2) 프랜차이즈 브랜드 입점 관리
	3) 프랜차이즈 브랜드 검색 및 조회
	4) 브랜드 가맹점 상세 정보(포트폴리오)
	5) 관심 브랜드 찜하기
	6) 창업 컨설팅 상담 예약
	7) 상담 신청 상태, 이력 관리 및 알림
	8) 임베딩 영상 스트리밍

3. 애플리케이션 설계 및 구조:

3.1. DB 모델링(ERD)



User (회원)

컬럼명	타입	제약조건	설명
user_id	bigint	PK	사용자 식별자
email	varchar(255)	UNIQUE, NOT NULL	로그인 계정
password	varchar(255)	NOT NULL	암호화 저장
name	varchar(100)	NOT NULL	사용자 이름
phone	varchar(20)		연락처

BrandManager (브랜드 매니저)

컬럼명	타입	제약조건	설명
manager_id	bigint	PK	브랜드 매니저 식별자
name	varchar(100)	NOT NULL	매니저 이름
password	varchar(255)	NOT NULL	암호화 저장
email	varchar(255)	NOT NULL	매니저 이메일
phone	varchar(20)	NOT NULL	연락처

BrandCategory (브랜드 카테고리)

컬럼명	타입	제약조건	설명
category_id	bigint	PK	카테고리 식별자
category_name	varchar(50)	UNIQUE	외식, 교육, 뷰티, 편의점

Brand (브랜드)

컬럼명	타입	제약조건	설명
brand_id	bigint	PK	브랜드 식별자
brand_name	varchar(200)	NOT NULL	브랜드명
category_id	bigint	FK → BrandCategory.category_id	브랜드 카테고리
manager_id	bigint	FK → BrandManager.manager_id	담당 매니저

BrandDetail (브랜드 상세정보)

컬럼명	타입	제약조건	설명
detail_id	bigint	PK	상세정보 식별자
brand_id	bigint	FK → Brand.brand_id	브랜드
view_count	bigint	DEFAULT 0	조회수
save_count	bigint	DEFAULT 0	저장수
initial_cost	decimal(15,2)	NOT NULL	가맹비
total_investment	decimal(15,2)	NOT NULL	평균 총 창업비용
avg_monthly_revenue	decimal(15,2)	NOT NULL	평균 월매출
store_count	int	NOT NULL	전체 매장수
brand_description	varchar(200)	NOT NULL	브랜드 한줄설명

SavedBrand (관심 브랜드 저장)

컬럼명	타입	제약조건	설명
save_id	bigint	PK	저장 식별자
user_id	bigint	FK → User.user_id	사용자
brand_id	bigint	FK → Brand.brand_id	브랜드

saved_at	timestamp	DEFAULT now()	저장 시각
----------	-----------	---------------	-------

정책: (user\_id, brand\_id) Unique 제약 필요

#### ConsultationStatus (상담 상태 코드)

컬럼명	타입	제약조건	설명
status_code	varchar(20)	PK	상담 상태 코드
status_name	varchar(50)		상태명

- 상태 값:
  - PENDING: 신청 중
  - RESCHEDULE\_REQUEST: 조정 요청
  - CONFIRMED: 확정
  - CANCELLED: 취소

#### Consultation (상담 신청)

컬럼명	타입	제약조건	설명
consultation_id	bigint	PK	상담 식별자
user_id	bigint	FK → User.user_id	신청자
brand_id	bigint	FK → Brand.brand_id	브랜드
status_id	bigint	FK → ConsultationStatus.status_code	상담 상태
preferred_date	date		희망 상담일
preferred_time	time		희망 상담시간
manager_note	text		매니저 메모
created_at	timestamp	DEFAULT now()	신청 시각
confirmed_at	timestamp		상담 확정 시각
completed_at	timestamp		상담 완료 시각

#### Notification (알림)

컬럼명	타입	제약조건	설명
notification_id	bigint	PK	알림 식별자
consultation_id	bigint	FK → Consultation.consultation_id	관련 상담

message	varchar(200)		알림 메시지
created_at	timestamp	DEFAULT now()	알림 발생 시각
is_read	boolean	DEFAULT false	읽음 여부

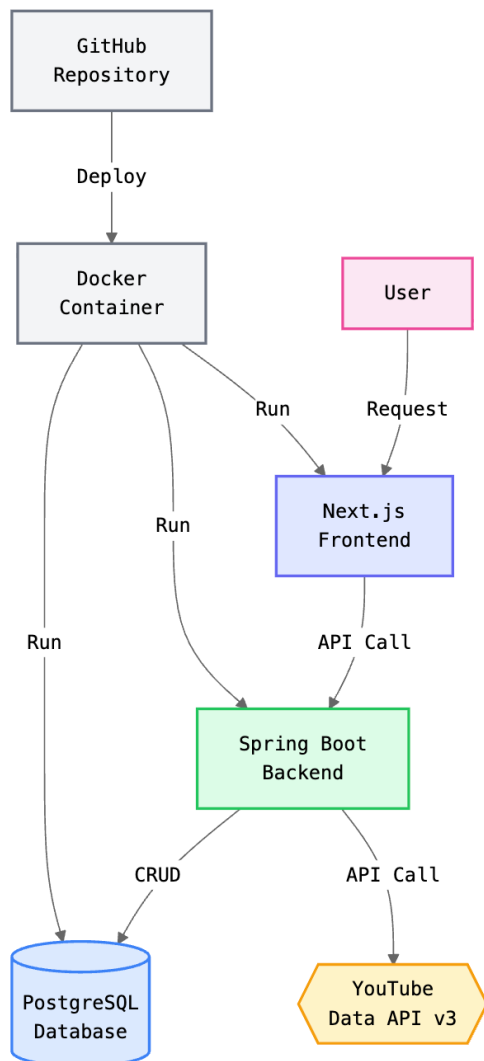
## 3.2. 백엔드/프론트엔드 구조

### 3.2.1. 기술 스택

백엔드: Spring Boot, PostgreSQL, Docker

프론트엔드: Next.js

### 3.2.2. 아키텍처



단, 콘텐츠 저작권 등의 문제로 어떤 TV 스트리밍 API도 무료/저가로 제공되고 있지 않아 실제 TV 연동이 불가하였다. 대안으로 YouTube Player API를 연동하여 임베드, 콘텐츠 재생 스크린으로 활용하였다.

### **3.3. 기능 시연:**

화면 녹화 동영상으로 대체한다.

## **4. 소감**

### **4.1. 프로젝트를 진행하며 배운 점**

#### **4.1.1. 개인적 성장**

본 프로젝트를 통해 풀스택 개발을 경험했다는 의의가 있다. 이전에는 해커톤에서 프론트엔드만 담당하거나, 데이터베이스 수업에서 DB 구조 설계만 진행하는 등 주로 단편적인 과제들을 다루는 데 그쳤다. 일주일 동안 엔티티 설계부터 BE 구현, FE 구현까지 차근차근 구축해나가는 과정에서 전반적인 개발 프로세스에 대한 이해도를 크게 향상시킬 수 있었다. JAVA와 SPRING BOOT 스택을 처음 경험하며 숙련도를 높인 것도 큰 성장의 요소이다.

#### **4.1.2. AI 도구 활용 역량 향상**

AI 도구를 활용하더라도 릴레이션 관계와 주요 서비스 로직을 개발자가 직접 설계하고 이해하고 있을 때, 코드가 스파게티화되지 않고 의도한 구조대로 안정적으로 구현됨을 확인했다. 오류 발생 시에도 근본 원인을 파악하고자 하는 태도를 배웠고, 실제로도 코드 구조나 에러의 근본 원인 등을 공부한 뒤 커서 등의 AI를 활용했을 때, 보다 효과적으로 오류를 해결할 수 있었다.

#### **4.1.3. 프롬프트 엔지니어링 역량 향상**

프로젝트 진행 과정에서 다음과 같은 AI 프롬프팅 기법을 습득했다:

- 프론트엔드는 백엔드 구조 완성 후 이를 표출하기에 가장 적합한 형태로 구현 요청 시 최적화된 결과가 도출된다.

- 이 때, 참조 이미지 첨부을 통하여 UI 구현 정확도를 향상할 수 있다.
- 컨텍스트는 최대한 좁힐수록 정확도와 응답 품질이 개선된다.
- 오류 발생 시 문제 영역을 직접 분석하여 원인 후보를 리스트업한 후, 해당 내용을 프롬프트에 포함하여 구체적으로 검토를 요청하면 더 빠르게 더 정확한 수정 작업을 진행할 수 있다. 예를 들어, 에러코드 500 발생 시 단순 500 이 발생하니 해결해 달라고 프롬프팅하는 것이 아니라 엔티티와 api 요구사항 매핑에 문제가 있거나/페이지네이션 로직에서 dto 처리에 문제가 있거나/로그인 세션 유지 코드에서 헤더 받아오는 부분에 문제가 있을 것이라는 식으로, 의심 가는 부분을 분석해서 프롬프팅했을 때 가장 오래걸리면서도 가장 빠른 방법이 된다.
- Ask 모드를 통하여 문제 진단 및 해결 프로세스 계획 수립을 시킨 후, 타당한 답변을 내놓으면 Agent 모드로 실행하는 방식으로 단계적으로 접근하는 것이 효율적이다.

## 4.2. 아쉬웠던 점

### 4.2.1. 기술적 한계

관련 API 의 접근성에 제약이 있어서 YouTube API 를 활용하는 방향으로 대체하였는데, 실시간 스트리밍 영상 데이터 처리 경험을 하지 못한 점은 아쉽다.

### 4.2.2. 시간 관리

개발 시 발생하는 각종 에러 사항을 해결하는 데 분배한 시간이 많아, 전체 프로젝트 일정에 영향을 미쳤다.

### 4.2.3. 프로젝트 지속성

노력과 시간을 투입한 만큼 기획 단계에서부터 프로젝트 완료 후에도 지속적으로 고도화하여 실용적인 서비스로 발전시킬 수 기회가 있으면 더 좋겠다는 아쉬움이 있다.