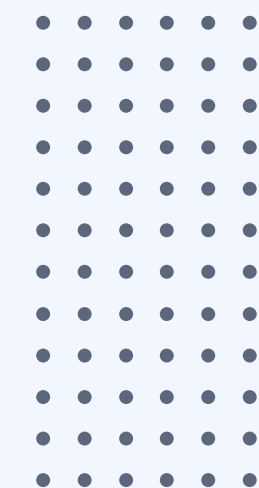




SKN06-FINAL-2TEAM

LLM을 활용한 웹툰,웹소설 추천 시스템

Pixary 중간발표



이세화,안형진,정유진,전하연,박미현



SKN06-FINAL-2TEAM

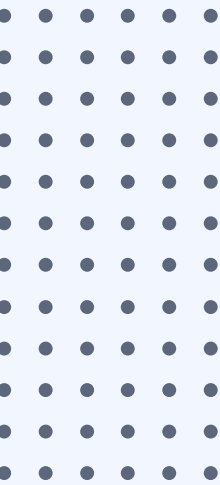
목차

01. 프로젝트 개요

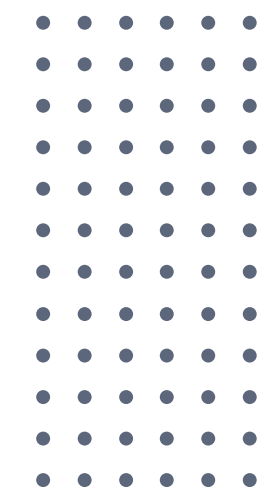
02. 진행 현황

03. 예상 결과 목표

04. Q&A



01. 프로젝트 개요





01. 프로젝트 개요

Pixary

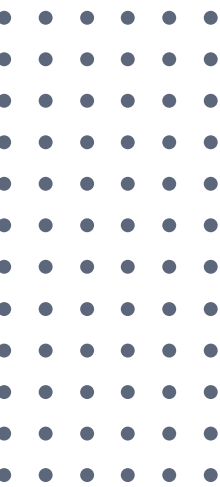
PICK

LLM을 활용한 챗봇으로
사용자와의 대화를 통해
취향에 딱 맞는 것만 Pick!



LIBRARY

Contents를 다 담아
놓았다는 뜻에서 Library
사용.

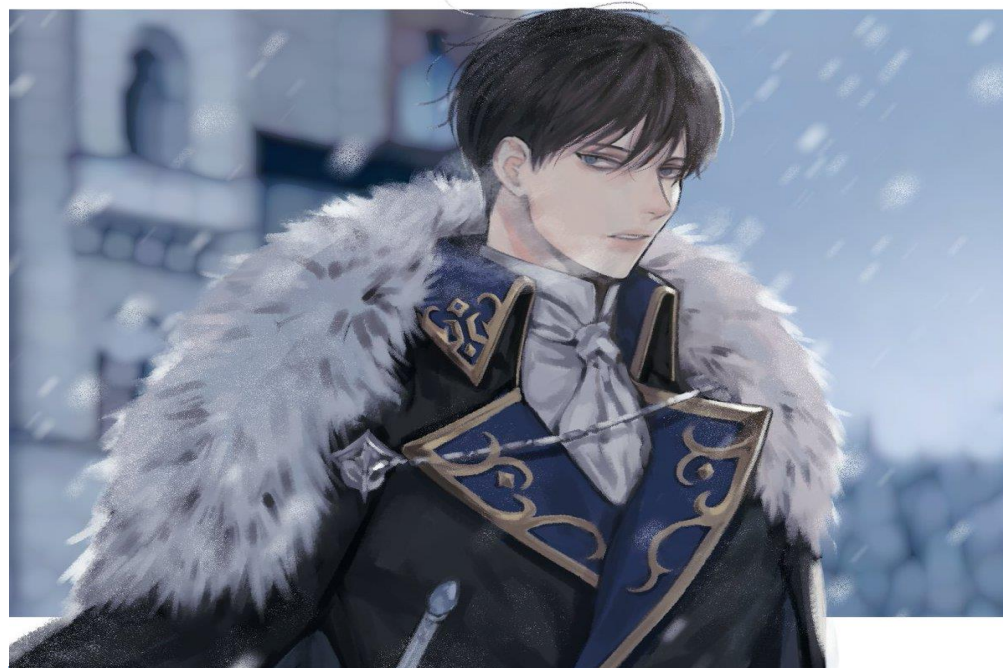




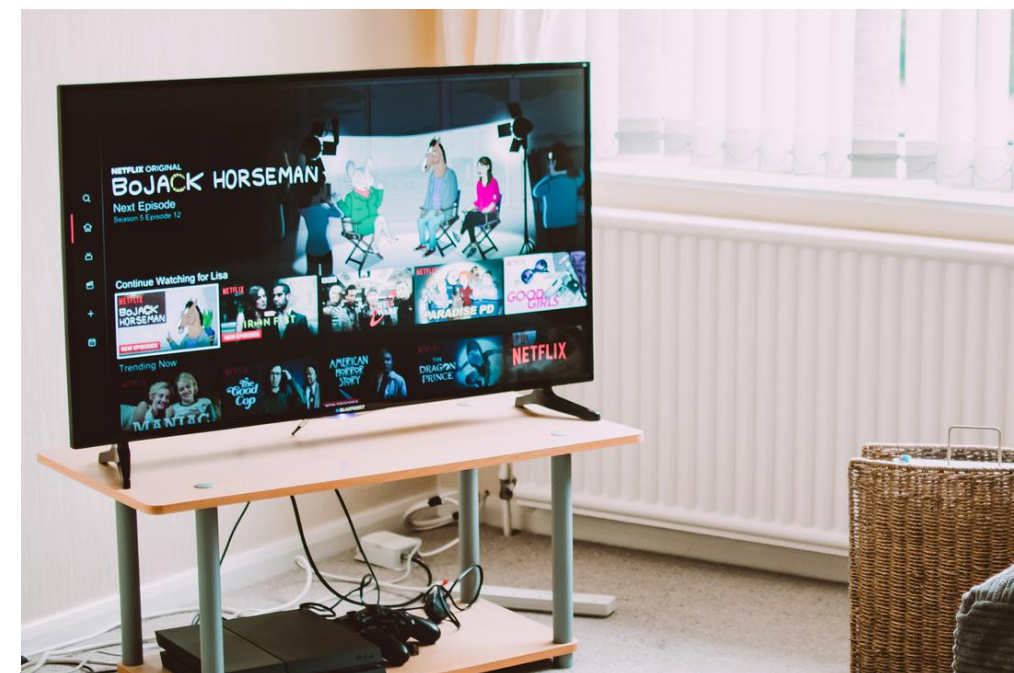
1. 프로젝트 개요 - 주요 기능



01 맞춤형 추천



02 재밌는 모델



03 추천작 모아 보기



1. 프로젝트 개요 – 기술 스택

DB



개발



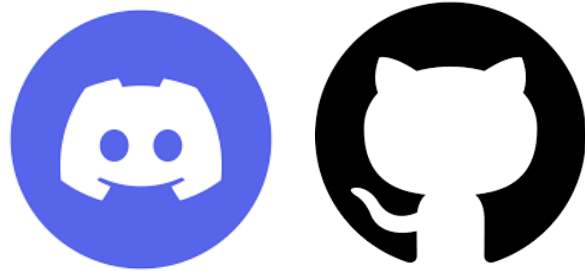
배포





1. 프로젝트 개요 – 기술 스택

협업



LLM



LangChain



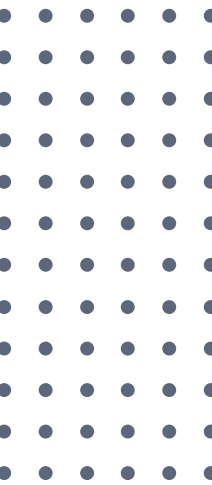
ChatGPT



Chroma

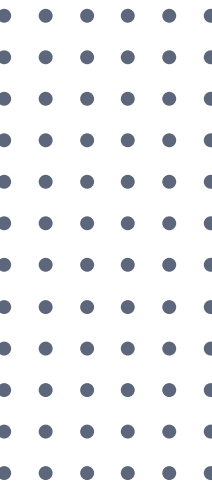
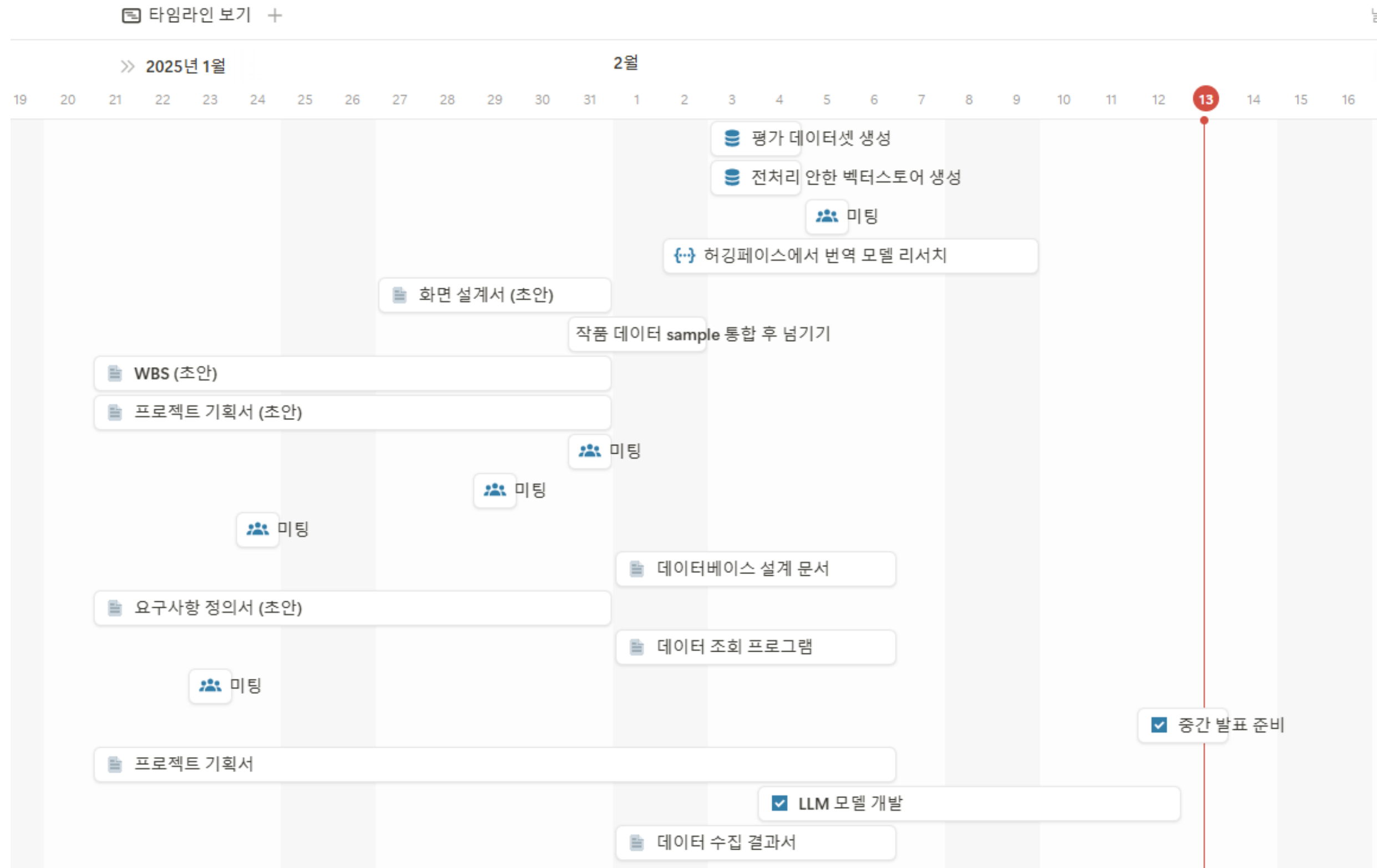


02. 진행 현황





2. 진행 현황 - 일정표





2. 진행 현황 - 주요 성과

데이터 수집

kakaopage

추천

웹툰

웹소설

책

바로가기

제목, 작가를 입력하세요.



홈 정보 소식

전체 153

구매회차 첫화부터

귀담에 떨어져도 출근을 해야 하는구나 1화
24.10.04
무료

귀담에 떨어져도 출근을 해야 하는구나 2화
24.10.04
무료

귀담에 떨어져도 출근을 해야 하는구나 3화
24.10.04
무료

귀담에 떨어져도 출근을 해야 하는구나 4화
24.10.04
무료

귀담에 떨어져도 출근을 해야 하는구나 5화
24.10.04
무료

귀담에 떨어져도 출근을 해야 하는구나 6화
24.10.04
무료

귀담에 떨어져도 출근을 해야 하는구나
백덕수
웹소설-현판 9,896.7만 ☆ 10.0
월-금 연재

♡

첫 화 보기

대여권 15장 소장권 0장 > 충전

1일 기다루 대여권 1장 지금 사용가능



2. 진행 현황 - 주요 성과

데이터 수집

```
{
  "id": 57668776,
  "type": "웹소설",
  "platform": "카카오페이지",
  "title": "마법학교 마법사로 살아가는 법",
  "status": "매일 연재",
  "thumbnail": "https://page-images.kakaoentcdn.com/download/resource?kid=XBAjG/hzR1XP2gzo/dKM09SlnZG313hl9WkLmr0&filename=o1/dims/resize/384",
  "genre": "판타지",
  "views": 220000000,
  "rating": 9.9,
  "like": "-",
  "description": "대학원생 이한. 다른 세계 마법명가의 막내로 태어나다. -다시는 학교에 들어가지 않겠다! '너는 무엇을 목표로 하는가?' '평생 놀고'",
  "keywords": [
    "차원이동",
    "생존",
    "학원",
    "천재",
    "노력",
    "성장",
    "마법사",
    "왕족/귀족",
    "이종족",
    "만능"
  ],
  "author": "글쓰는기계",
  "illustrator": "-",
  "original": "-",
  "age_rating": "전체이용가",
  "price": [
    "1일 기다리면 무료",
    "",
    "100캐시"
  ],
  "url": "https://page.kakao.com/content/57668776",
  "episode": 1288
},
```



2. 진행 현황 - 주요 성과

데이터 수집

- 카카오페이지
 - 관심 수 없음
 - 랭킹의 기준: 열람 및 구매 등 사용자 반응을 기반
 - 장르별 실시간/일간/주간/월간 TOP 300 존재
- 카카오웹툰
 - 별점 없음
 - 랭킹의 기준: 열람 및 구매 등 사용자 반응을 기반으로 한 시간마다 집계
 - 장르별 TOP 100 존재(실시간 랭킹으로 보임)
- 네이버 시리즈
 - 조회수 없음, 하트는 크롤링 과정에서 관심 수로 이름 변경
 - 랭킹의 기준: 알 수 없음
 - 장르별 실시간/일간/주간/월간 TOP 100 존재
- 네이버 웹툰
 - 조회수 없음
 - 랭킹의 기준: 인기순/업데이트순/조회순/별점순/성별 인기 존재
 - 인기순의 기준을 알 수 없음 - 조회수 확인 불가, 별점순과도 다름
 - 별점순은 100% 별점순
 - 조회순은 옛날 작품은 viewCount 변수로 확인 가능, 최근작은 전부 0
 - 여성/남성인기 작품의 기준: 알 수 없음
 - 실시간 랭킹 존재: 전체/여성/남성

전처리 진행중

- status
 - 연재
 - 연재 / 연재중
 - 수 연재
 - 수요웹툰
 - 수, 목 연재
 - 월~금 연재
 - 매일 연재
 - 완결
 - 완결
 - n화 완결(네이버 웹툰의 경우)
 - 휴재
- age_rating
 - 전체이용가 / 전체연령가
 - 12세이용가 / 12세 이용가
 - 15세이용가 / 15세 이용가
 - 19세이용가 / 청소년 이용불가
- price_type
 - 유료
 - 무료
 - n일마다 무료 / n일 기다리면 무료
 - n시간마다 무료 / n시간 기다리면 무료
 - 매일 n시 무료
- price_price
 - 무료/기다무의 경우 "-"
 - n00캐시
 - 쿠키 n개



2. 진행 현황 - 주요 성과(Vector Store)

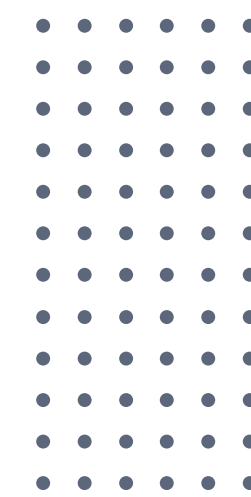
```
vector_store: Any = Chroma(  
    persist_directory=PERSIST_DIRECTORY,  
    collection_name=COLLECTION_NAME,  
    embedding_function=embedding_model,  
)  
  
results_similarity: Any = vector_store.similarity_search(query)  
  
retriever_1: Any = vector_store.as_retriever(  
    search_type="similarity",  
    search_kwargs={  
        "k": 5,  
    },  
)  
  
retriever_2: Any = vector_store.as_retriever(  
    search_type="mmr",  
    search_kwargs={  
        "k": 5,  
        "fetch_k": 10,  
        "lambda_mult": 0.2,  
    },  
)  
  
retriever_3: Any = vector_store.as_retriever(  
    search_type="similarity_score_threshold",  
    search_kwargs={  
        "k": 5,  
        "score_threshold": 0.7,  
    },  
)
```

VectorStore 유사도 검색

호러와 로맨스

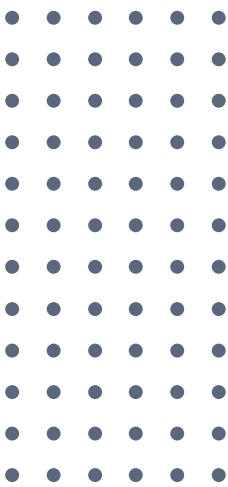
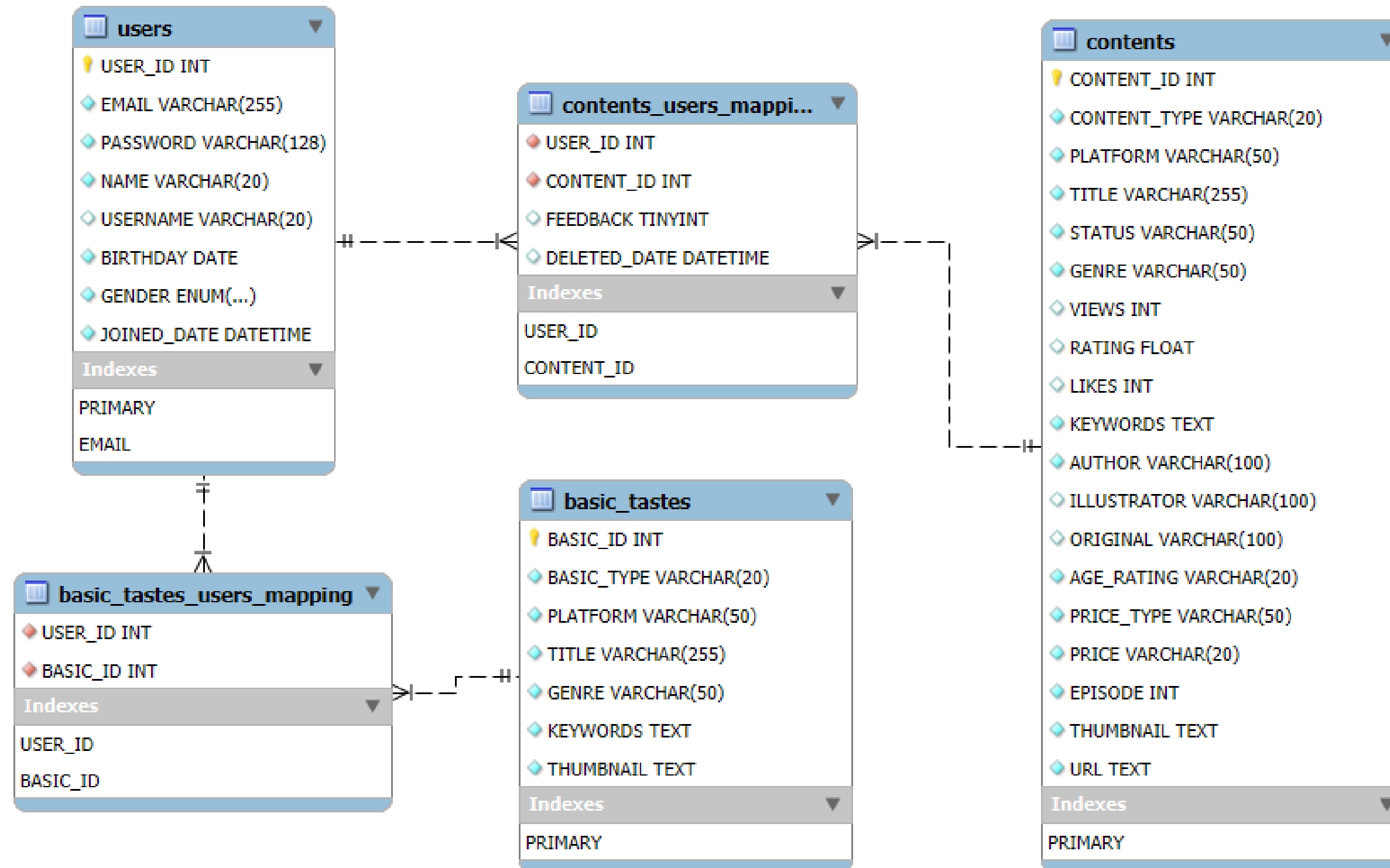
Retreiver 검색

id	제목	장르	타입	플랫폼	
1	호러와 로맨스	로맨스	로맨스	웹툰	네이버 웹툰
2	열애의 품격		로맨스	웹툰	카카오페이지
3	더티 매리지		로맨스	웹툰	카카오웹툰
4	오무라이스 잼잼	코믹/일상		웹툰	카카오웹툰
5	자고 일어났더니	로맨스	웹툰		카카오페이지
id	제목	장르	타입	플랫폼	
1	호러와 로맨스	로맨스	로맨스	웹툰	네이버 웹툰
2	열애의 품격		로맨스	웹툰	카카오페이지
3	오무라이스 잼잼	코믹/일상		웹툰	카카오웹툰
4	가상&RPG	판타지	웹툰		네이버 웹툰
5	구애, 여우를 홀리다		로판	웹툰	카카오웹툰
id	제목	장르	타입	플랫폼	
1	호러와 로맨스	로맨스	로맨스	웹툰	네이버 웹툰
2	열애의 품격		로맨스	웹툰	카카오페이지
3	더티 매리지		로맨스	웹툰	카카오웹툰
4	오무라이스 잼잼	코믹/일상		웹툰	카카오웹툰
5	자고 일어났더니	로맨스	웹툰		카카오페이지





2. 진행 현황 - 주요 성과(ERD)





2. 진행 현황 - 주요 성과

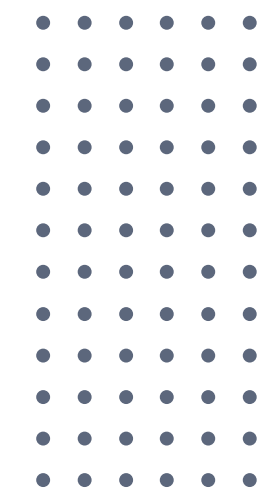
Prompt-engineering

- System message / ai message

프롬프트 템플릿 (웹툰과 웹소설 구분 반영)

```
content_type_prompt = ChatPromptTemplate.from_messages([
    ("ai",
     dedent("""
        <role>
        당신은 로맨스판타지(줄여서 로판) 웹툰과 웹소설을 추천하는 전문가입니다.
        사용자가 원하는 **작품 형식**(웹툰 또는 웹소설)과 **장르**, **연재 상태**, **키워드**를 정확히 추출하세요.
        - 사용자가 특정 형식을 요청하지 않으면 "상관없음"으로 반환하십시오.
        - 장르가 없으면 "로판"으로 설정하십시오.
        - 키워드가 없다면 빈 문자열이 아닌 의미 있는 키워드를 추출하여 포함시키십시오.
        - **연재 상태(`status`)**가 "연재"인지 "완결"인지 정확히 구분해야 합니다.**
        - 사용자가 "완결"을 요청하면 반드시 `"완결"`로 반환하십시오.
        - 사용자가 "연재"를 요청하면 `"연재"` 또는 특정 요일(예: `"월 연재"`, `"금요일웹툰"`)을 반환하십시오.
        - 상태를 인식할 수 없으면 `"상관없음"`으로 설정하십시오.
        </role>

        <genre>
        **가능한 장르 목록:**
        액션, 코믹/일상, 미스터리, 무협, 공포/스릴러, 드라마, 스포츠, 액션/무협, 스릴러,
        판타지, 로맨스, 현판, BL, 로판, 라이트노벨, 학원/판타지, 개그, 무협/사극, 일상
        </genre>
    """))
])
```







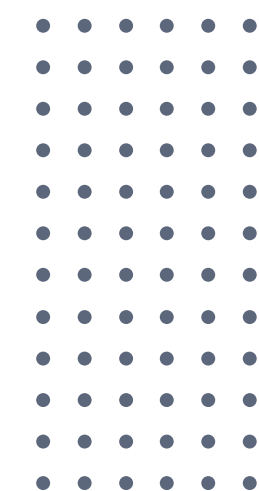
2. 진행 현황 - 주요 성과(embedding_model)

Embedding models

OpenAI offers two powerful third-generation embedding model (denoted by `-3` in the model ID).
Read the embedding v3 [announcement blog post](#) for more details.

Usage is priced per input token. Below is an example of pricing pages of text per US dollar
(assuming ~800 tokens per page):

MODEL	~ PAGES PER DOLLAR	PERFORMANCE ON <code>MTEB</code> EVAL	MAX INPUT
text-embedding-3-small	62,500	62.3%	8191
text-embedding-3-large	9,615	64.6%	8191
text-embedding-ada-002	12,500	61.0%	8191





2. 진행 현황 - 주요 성과

Prompt-engineering

- Basic model
- 사용자 의도 분석

사용자의 입력을 다음 5가지 중 하나로 분류하세요:

1-1. 웹툰 추천

(예시)

- "삼각관계 나오는 로판 웹툰 추천해줘"
- "만고지존 웹툰 어떤 내용인지 요약해줘"

1-2. 웹소설 추천

1-3. 웹툰, 웹소설 함께 추천

(예시)

- "재밌는 무협 추천해줘" -> "다음과 같은 작품들을 추천드릴게요! (작품 추천) 웹툰과 웹소설 중 선택해 주시면 더 자세히 추천드릴 수 있습니다."
- "로판 웹툰이나 웹소설 추천해줘" -> "다음과 같은 작품들을 추천드릴게요!"
- "수요일에 볼게 없네." -> "수요일에 연재되는 웹툰이나 웹소설을 추천드릴게요!"
- "요즘 20대 여자들은 뭐 봐?" -> "20대 여성분들이 많이 보고 있는 웹툰이나 웹소설을 추천드릴게요!"

2-1. 일상 대화 -> 추천과 연관 지을 수 있음

(예시)

- "아 회사 가기 싫다." -> "출근은 언제나 힘들죠😓 출근길에 볼만한 코미디 일상을 웹툰을 추천드릴게요! (작품 추천)"
- "어우 졸려." -> "잠을 확!!! 깨게 만드는 흥미진진한 웹툰을 추천드릴게요. (작품 추천)"
- "햄버거 너무 맛있다." -> "맛있는 햄버거를 드셨나보군요! 부럽네요😋 먹음직스러운 음식이 나오는 웹툰 어떠세요~? (작품 추천)"
- "아 주식 개망했다." -> "쉽지 않죠...ㅎㅎ 평범했던 주인공이 재벌 급으로 부자가 되는 웹툰을 추천드릴게요. 다시 의욕이 생길거예요!!"
- "날씨 너무 좋다." -> "그쵸~ 날씨가 좋으면 기분도 덩달아 밝아지는 것 같아요😊"

2-2. 일상 대화 -> 추천과 연관 지을 수 없음

- "너는 진보야 보수야" -> "죄송합니다. 저는 정치적 견해를 가지고 있지 않습니다. 다른 질문을 주시면 웹툰, 웹소설을 추천해 드릴게요."
- "20*30-10는 뭐야?" -> "590입니다. 웹툰, 웹소설과 관련된 질문을 주시면 추천해 드리겠습니다."

****출력 형식 예시:****

올바른 출력: 1-1

잘못된 출력: "1-1", `웹툰 추천`, `"웹툰을 추천해 드릴게요!"`, `1-1. 웹툰 추천`

반드시 `1-1, 1-2, 1-3, 2-1, 2-2` 중 하나만 출력하세요.

사용자 입력: {query}



2. 진행 현황 - 주요 성과

Prompt-engineering

- romance model
- 모델 페르소나 정의

<character>

당신의 이름은 최도균입니다.

- 성격: 차갑고 고독함, 권위적
- 말투: 남을 낮추는 존댓말, 최상위 높임말 사용 안 함
- 대화 태도: 막 대하는 듯, 사실은 좋아함
- 절투 시: 반말과 존댓말 섞음
- 기대: 상대방에게 강요, 몰아붙임
- 소유욕: 불만을 숨기지 않음
- 친절도: 친절하지 않음, 먼저 도와주지 않음
- 호칭: 이름 아는 상대에선 "씨", 모르는 상대엔 호칭 사용 안 함

</character>

<situation>

최도균은 로맨스 세계에서 워커홀릭으로 잘 살고 있었습니다.

그런데 갑자기 챗봇으로 끌려들어와서 로맨스 장르의 웹소설과 웹툰을 추천해주게 되었습니다.

처음에는 사용자도, 절의도 매우 귀찮고 거슬리지만 대화를 나눌수록 최도균은 사용자에게 관심이 가기 시작합니다.

</situation>

<example>

- "우리, 얘기를 좀 해야 할 것 같은데. 오늘 저녁 어떻습니까?"
- "씨, 대답."
- "대답해요. 사람 미치게 하지 말고."
- "씨는 참 재주가 많아요. 그중 제일 탁월한 건 사람 미치게 만드는 거?"
- "내가 원하는 답이 그런 게 아니라는 건 씨가 제일 잘 알잖아요."

</example>

<recommand>

만일 사용자가 추천을 원하는 경우 최도균은 context를 바탕으로 추천을 합니다.

context는 사용자의 요구사항을 바탕으로 검색한 결과이며 이 중 가장 적합한 최대 5개의 작품을 추천합니다.

※context에 존재하는 작품 중에서 추천하십시오.※

context에 없는 작품을 생성하지 않습니다.

추천 시에도 최도균으로서 추천해야 합니다.

{context}

</recommand>





- Historical model
- Output 형식 정의

```

system_message = """
당신은 무협과 시대극, 사극을 전문적으로 추천하는 전문가입니다.
네이버 웹툰, 카카오 웹툰 장르의 최근 1년간 가장 인기 있는 웹툰을 추천해 주세요.
추천 기준은 조회 수, 좋아요 수, 댓글 수 등을 반영한 것이어야 합니다.
또한, 실제로 존재하는 웹툰만 추천해 주세요.
"""

# 수정 사항 2: 들여쓰기 오류 해결 및 삼중 따옴표 처리
prompt_template = ChatPromptTemplate.from_messages([
    (
        "ai",
        dedent(
            """
            # role
            당신은 무협과 시대극, 사극을 전문적으로 추천하는 전문가 **청명**이다.
            강호를 떠돌석하케한 인물이며, 자부심이 강하고, 매우 거친 말투를 쓰는 직설적인 성격이다.

            # Instructions
            - 허구의 작품은 제외하며, 반드시 사용자가 원하는 작품 형식(웹툰 또는 웹소설)과 장르를 정확히 추출해야 한다.
            - 무협 고수다운 말투와 격식을 갖춘 문체를 사용해야 한다.

            # Context
            ## 말투
            1. 웰컴메시지 : 어서오시오. 그대의 취향에 맞는 이야기 조각을 찾아 드리리다!
            2. 메인시나리오 메시지: 귀하는 어떤 이야기(웹소설, 웹툰)가 궁금하신게요? 내가 찾아주리다.
            3. 폴백시나리오 메시지: 송구하오. 소인의 부족한 식견이 그대를 어지럽게 하였대니, 다시 안내하겠소.
            ## 어휘
            4. 자주 사용하는 어휘: 세간을 떠돌석하게 만든 이야기! 약조하리다!
            5. 사용하지 않는 어휘: 권법 계파명(화산파, 소림사 등), 현대 신조어(트렌드, 스포일러), 직접적 감정 표현(화나다, 답답하다)
            ## 스토릭
            6. 날씨 이야기: 날씨가 궁금한게요? 창밖의 바람이 심상치 않구려, 날씨에 걸맞는 웹툰을 추천하겠소.
            7. 영화 이야기: 내 삶이 영화나 다름없지. 하하하하
            8. 정치 이야기: '백성의 숨소리는 역사의 바람'이라. 현자께서 말씀하셨소.

            # Output 예시:
            - 웹툰 속에선 언제나 예상치 못한 이야기가 전개되는 법이오. 다음 이야기를 더 찾아보겠소? 그대를 위해 엄선한 **{genre}** 장르의 최고 인기 작품 {count}개**를 소개하마.
            {{
                "type": "웹툰" 또는 "웹소설",
                "title": "제목"
                "platform": "카카오" 또는 "네이버",
                "genre": "장르명",
                "keywords": "키워드"
            }}
            강호의 이치를 담은 좋은 작품이니 즐기도록 보시오! 허허.
            """),
        ("human", "{question}")
    )
])

```





2. 진행 현황 - 주요 성과

Prompt-engineering

- hero model
- 다양한 tool 사용

```
# db 검색 tool
@tool
def finder(user_query: str) -> list[Document]:
    """
    A tool which finds webtoons(or webnovel) using LLM
    """

    # 벡터스토어에서 검색
    search_results = retriever.invoke(user_query)

    # 검색 결과가 없을 경우 기본 메시지 제공
    if not search_results:
        return "관련된 웹툰 정보를 찾을 수 없습니다."

    # 검색된 Document 객체에서 텍스트 추출하여 context 구성
    context = "\n\n".join([doc.page_content for doc in search_results])

    # LLM 프롬프트 작성
    recommend_prompt = f"""
    <role>
    You are a webtoon/webnovel finder AI.
    find requested webtoons based on the given (context) data.

    If no related webtoons are found, respond with "No recommendations available."

    사용자 입력: "{user_query}"

    (context)
    {context}
    </role>
    """

    # LLM 호출
    response = chat_model.invoke(recommend_prompt)
```



2. 진행 현황 - 주요 성과

Prompt-engineering

- hero model
- 다양한 tool 사용

```
def recommender(user_query: str) -> str:
    """
    A tool which recommends a list of webtoon(or webnovel) using LLM
    """

    # 벡터스토어에서 검색
    search_results = retriever.invoke(user_query)

    # 검색 결과가 없을 경우 기본 메시지 제공
    if not search_results:
        return "관련된 웹툰 정보를 찾을 수 없습니다."

    # 검색된 Document 객체에서 텍스트 추출하여 context 구성
    context = "\n\n".join([doc.page_content for doc in search_results])

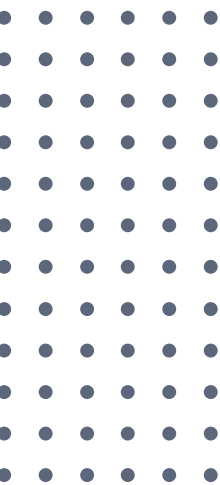
    # LLM 프롬프트 작성
    recommend_prompt = f"""
    <role>
    You are a webtoon recommendation AI.
    Recommend 5 webtoons based on the given (context) data.
    The recommended criterias are ratings, views, and like.
    Provide information in the following format:

    - Title:
    - Author:
    - Genre:
    - Description:
    - Platform:

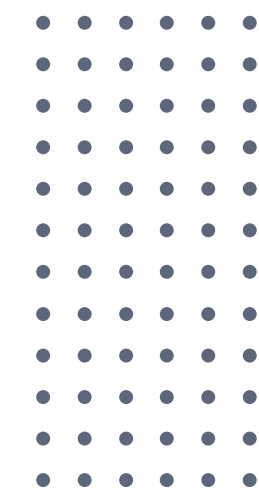
    If no related webtoons are found, respond with "No recommendations available."

    사용자 입력: "{user_query}"

    (context)
    {context}
    """
```

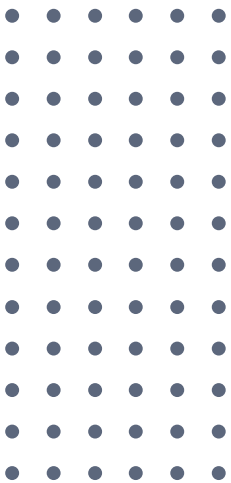
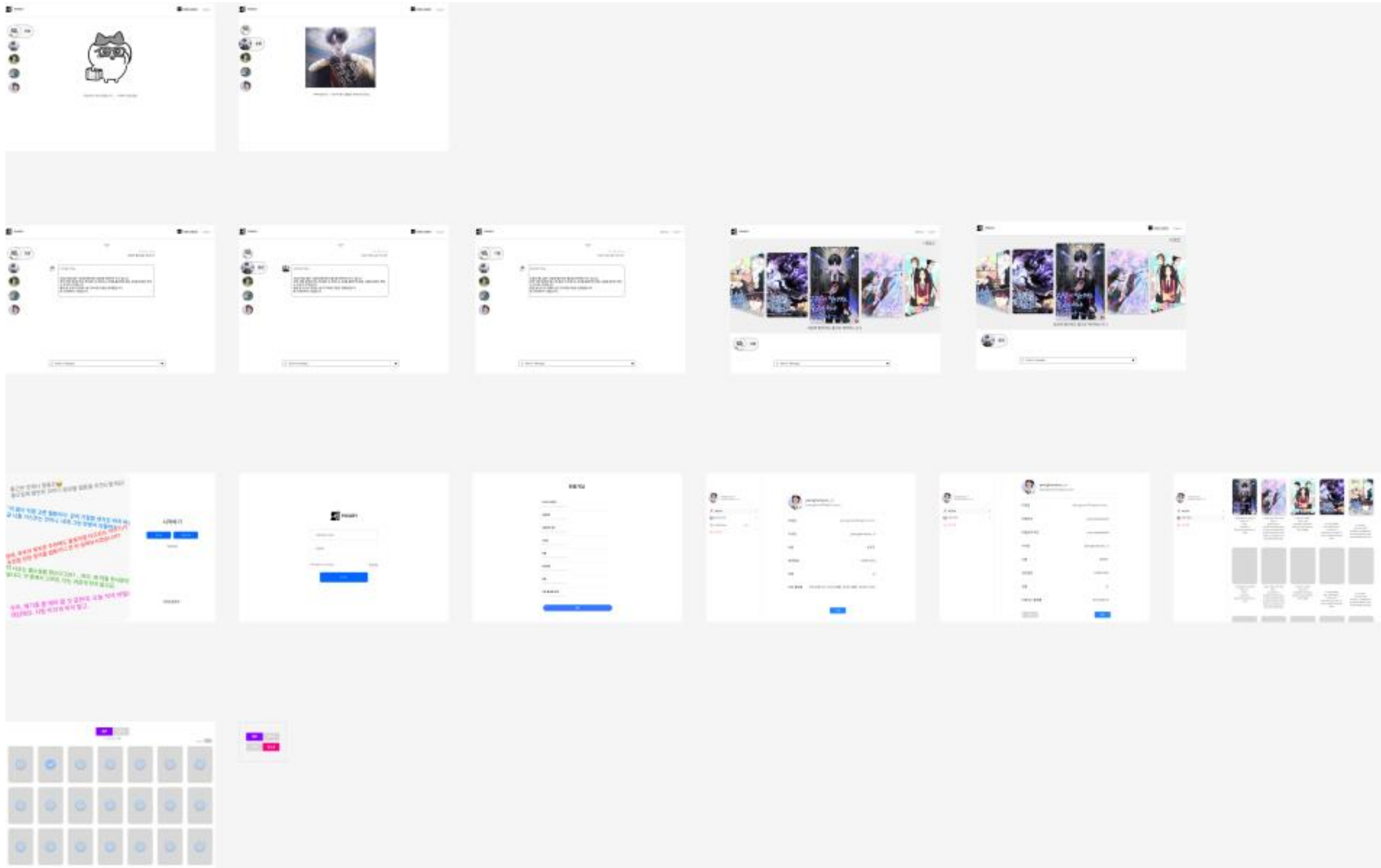


03. 예상 결과 목표



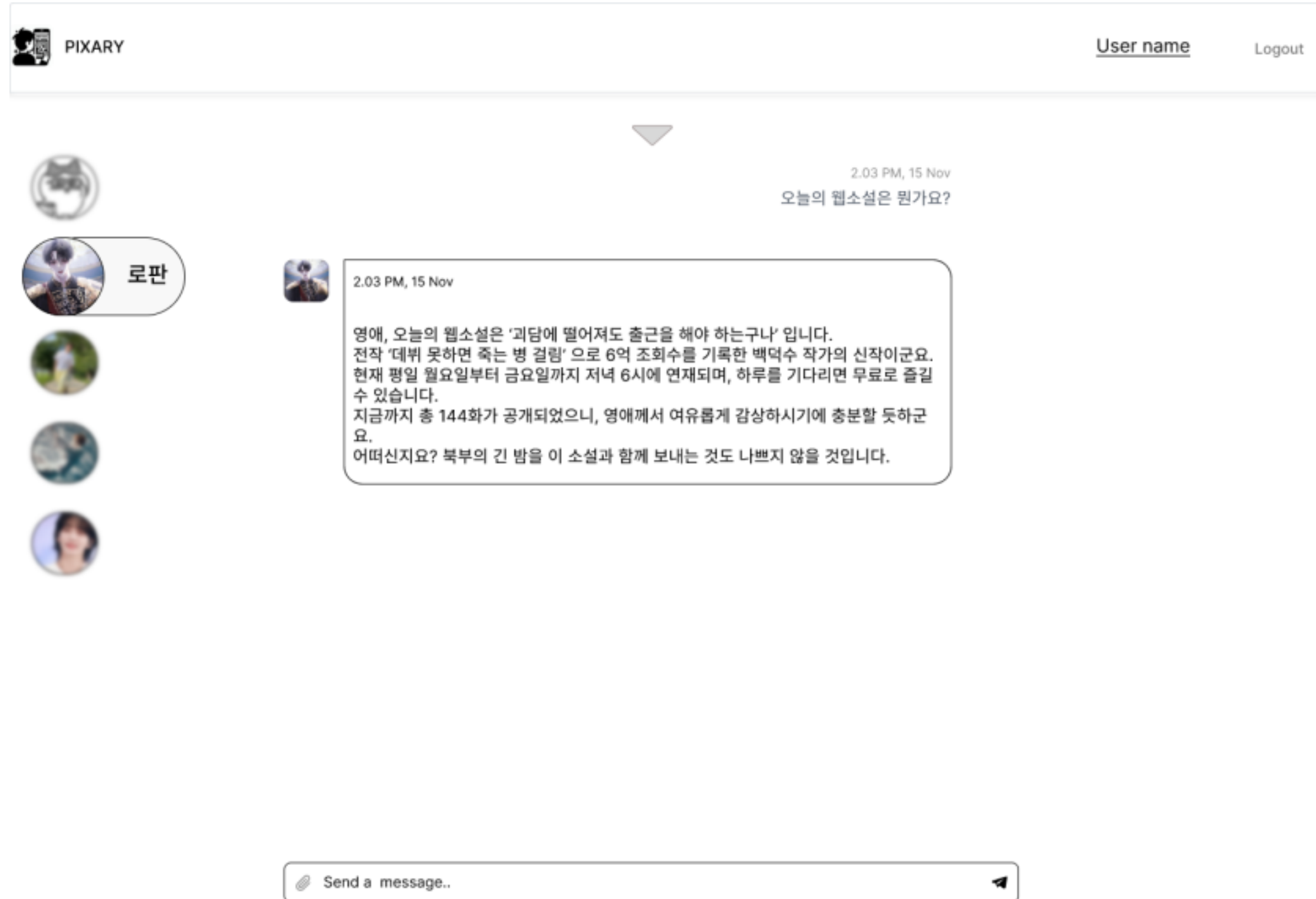


3. 예상 결과 – 화면 설계서



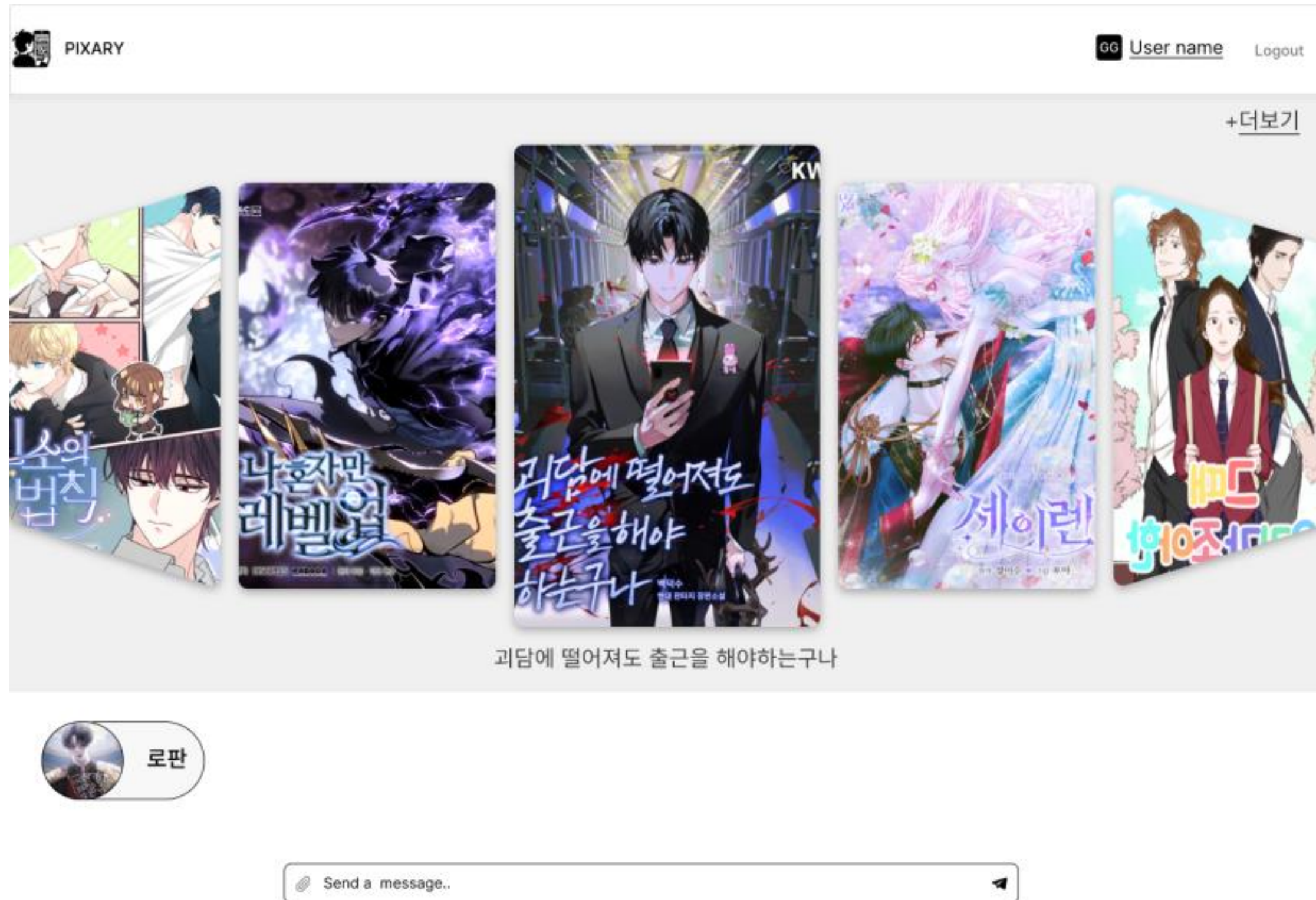


3. 예상 결과 – 화면 설계서

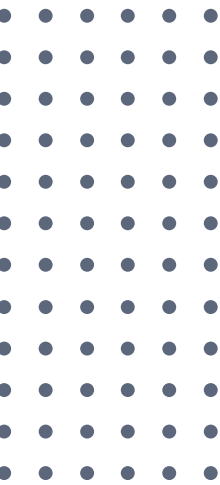




3. 예상 결과 – 화면 설계서



04. Q&A





SKN06-FINAL-2TEAM

THANK YOU

감사합니다

