

**[Toondy]**

**몰입감 넘치는 경험을 위한**

**웹툰 / 웹소설 BGM 생성 서비스**

**project plan**



Ver 1.2

팀명: 네모의 꿈

서영현

이예림

정재원

전홍석

Table of Contents

1. 프로젝트 제목 3

2. 필요성 및 배경 3

3. 목표 3

4. 팀별 역할 4

5. 요구사항 분석 및 정의 4

5.1 자료 조사 4

5.2 경쟁사 벤치마킹 4

5.3 요구사항 정의 5

6. 서비스 설계 6

6.1 서비스 개념 및 구조도 6

6.2 서비스 설명 6

6.2.1 Toondy(툰디) 서비스 개요 6

6.2.2 전체 서비스 구조 및 흐름 6

7. UX 설계 7

8. 데이터 설계 8

9. 프로그램 모듈 구조, 모델 및 알고리즘 설계 9

10. 테스트 계획 11

11. 구현 12

12. 결과 12

13. 감사의 글 13

14. 레퍼런스 13

# 프로젝트 제목

“**Toondy**”, 몰입감 있는 웹툰/ 웹소설 감상을 위한 AI BGM 제공 서비스 개발 기획

# 필요성 및 배경

AI가 미디어를 바꾸고 있는 시대, 우리는 웹툰·웹소설처럼 대중성과 콘텐츠 파급력이 높은 분야에 주목했습니다. 현재 웹툰과 웹소설은 시각적 요소와 텍스트 중심의 감상에 한정되어 있어 독자에게 단일 감각 기반의 콘텐츠 경험을 제공합니다. 작품의 분위기나 감정선은 독자의 해석에 전적으로 의존하는 경우가 많아 장면에 따라 몰입감이 저하될 수 있습니다.

또한 기존의 배경음악 서비스는 일부 작품에 한해 고정된 음원을 일괄적으로 삽입하거나 단순 추천 위주로 제공되고 있어 각 장면의 맥락이나 감정 흐름을 섬세하게 반영하는 데 한계가 존재합니다.

따라서 Toondy는 단순히 음악을 추천하는 것이 아니라 작품의 분위기와 장면 전개를 고려한 ‘맞춤형 생성’을 통해 독자의 감상 경험을 더욱 몰입감 있게 만들어줍니다. 이는 미디어와 생성형 AI의 접점에서 새로운 가치를 제시하고 콘텐츠 소비 방식에 변화를 이끌 차별화된 서비스입니다.

# 목표

Toondy의 목표는, 감성 기반 AI 기술을 통해 웹툰과 웹소설의 감상 방식을 ‘듣는 경험’으로 확장하고, 몰입형 미디어 시대를 선도하는 것입니다.

우리는 Toondy를 통해, 독자 한 사람 한 사람의 감정선에 맞춘 맞춤형 BGM을 생성함으로써, 기존 텍스트 콘텐츠를 감각적 미디어로 재해석하는 것을 목표로 합니다.

Toondy의 궁극적인 목표는, 사용자가 단순히 ‘읽는’ 것을 넘어, 장면마다 음악으로 ‘느끼고 몰입하는’ 새로운 콘텐츠 경험을 제공하는 것입니다.

# 팀별 역할

각 역할을 다음과 같이 기술. 모두가 동일한 역할(예. 개발)이라면, 맡은 각 프로그램 모듈별로 역할을 구분해야 합니다(모듈A 설계개발, 모듈B 설계개발 등).

|  |  |  |
| --- | --- | --- |
| 이름 | 역할 | 산출물 |
| 서영현 | 조장, 코드 통합, 배포 | Hugging Face 배포, Git, 개발 |
| 이예림 | 기획, 코드 기반 개발 | 시연 영상, Git, 개발, 기획서 |
| 정재원 | 모듈B 개발2 | 데이터 설계, 웹 크롤링 |
| 전홍석 | 개발 및 테스트 | UX 설계, 기획서, 모델 개발 |

# 요구사항 분석 및 정의

## 자료 조사

웹툰/웹소설 시장은 최근 몇 년간 모바일 콘텐츠 소비 증가와 함께 급성장 중이며   
감성 기반의 사용자 경험(UX)에 대한 수요가 확대되고 있음.

음악 생성 AI 기술(예: MusicGen, Riffusion 등)의 발전으로 정적인 콘텐츠에 동적인 오디오 요소를 결합하는 시도가 활발하게 이루어지는 추세.

LLM 기반 감정 분석 기술을 활용한 **멀티모달 AI 콘텐츠 서비스**에 대한 관심이 높아지고 있으며 향후 인터랙티브 콘텐츠 확장 가능성이 주목받고 있음.

## 경쟁사 벤치마킹

**\* BGM 관련 AI/콘텐츠 사례  
Riffusion**: 텍스트 프롬프트 기반 음악 생성. 정교한 사운드보다는 접근성과 창의성 중심.  
**Endel**: 상황·감정 기반 환경 음악 생성. 수면, 집중 등 맥락 기반 자동 음악 생성에 특화.  
**Webtoon 플랫폼 (ex. 네이버/카카오 웹툰)**: 특정 에피소드에 배경음 삽입 기능 있으나, **사용자 맞춤형 또는 자동 생성 기능은 없음.**

**\* 벤치마킹 포인트  
기존은 ‘고정 음악’ 중심**이며, 장면 맥락에 따라 음악이 실시간으로 변화하는 경험은 미흡함.  
**자동 생성형 음악**과 **스토리 감정선 연동**이라는 핵심 기능은 경쟁사 대비 차별화 지점.

## 요구사항 정의

주 사용자(고객)는 다음과 같습니다.

1. **웹툰·웹소설을 자주 감상하는 Z세대~MZ세대 독자**

– 감성적 몰입, 감각적 콘텐츠 소비를 선호

– 새로운 경험에 열려 있으며 콘텐츠 감상 UX에 민감

2. **웹툰/웹소설 창작자 및 플랫폼 서비스 기획자**

– 독자의 체류 시간 및 몰입도를 높이기 위한 기능적 차별화 니즈 존재

– 창작 과정에서 감정선에 맞춘 사운드 디자인 도입을 고려할 수 있음

요구사항은 다음과 같습니다.

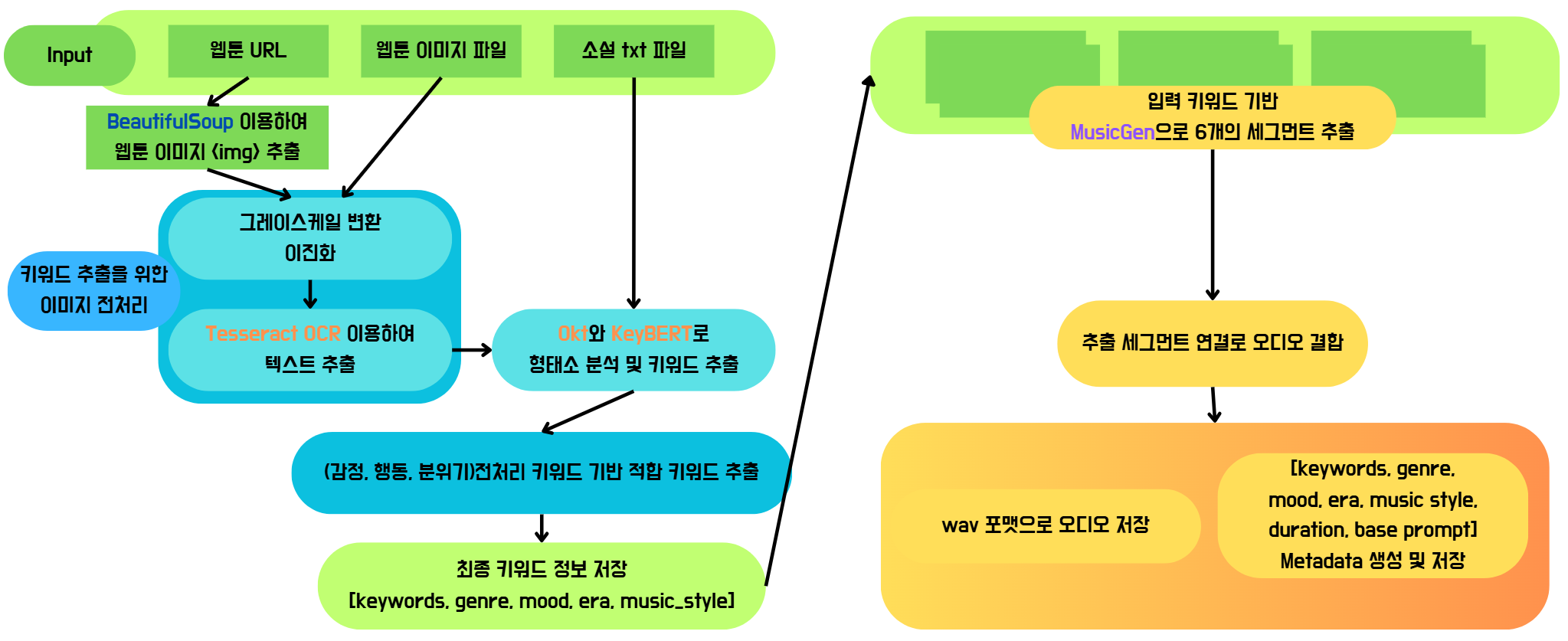
1. 텍스트 기반 콘텐츠(웹툰/웹소설)의 감정 및 전개 흐름을 분석할 수 있는   
AI 모델을 구축해야 합니다.

2. 분석된 감정과 장면 흐름에 맞춰 BGM을생성하고 콘텐츠와 실시간 동기화되는 감상 환경을 구현해야 합니다.

3. 사용자는 감상 중 음악을 끄거나 전환할 수 있으며 맞춤화 설정(장르/분위기 선택 등)이 가능한 UI가 제공되어야 합니다.

# 서비스 설계

## 서비스 개념 및 구조도



## 서비스 설명

### Toondy(툰디) 서비스 개요

Toondy는 웹툰 및 웹소설과 같은 텍스트 기반 콘텐츠의 감정 및 전개 흐름을 실시간 분석하고, 해당 감정에 동기화된 BGM을 자동으로 생성·재생하는 AI 기반 감상 지원 플랫폼입니다.  
 사용자의 몰입 경험을 극대화하기 위해, 콘텐츠의 내재된 감정/행동/분위기 데이터를 AI가 분석하고, 이를 기반으로 동기화된 음악을 제공하는 것이 핵심입니다.

### 전체 서비스 구조 및 흐름

**1. Input (입력 콘텐츠 수집)**

사용자가 감상할 웹툰 URL, 이미지 파일, 소설 텍스트 파일(txt) 등을 입력합니다.  
BeautifulSoup 등을 활용하여 웹상의 이미지 요소(img tag)를 추출하고, 분석을 위한 이미지 전처리를 수행합니다.

**2. 텍스트 추출 및 전처리**

웹툰 이미지에서 텍스트를 인식하기 위해 **Tesseract OCR**을 활용하여 말풍선 또는 내레이션 텍스트를 추출합니다.  
소설 등 txt 기반 입력은 전처리 없이 바로 형태소 분석으로 진입합니다.

**3. 감정 및 키워드 분석**

형태소 분석 도구인 **Okt 및 KeyBERT**를 활용하여 텍스트 내 키워드 및 핵심 문장들을 추출합니다.  
분석된 결과를 기반으로 감정(emotion), 행동(action), 분위기(mood) 를 파악하며, 이를 종합하여 음악 생성에 필요한 [keywords, genre, mood, era, music\_style] 데이터를 구성합니다.

**4. 음악 시퀀스 생성**

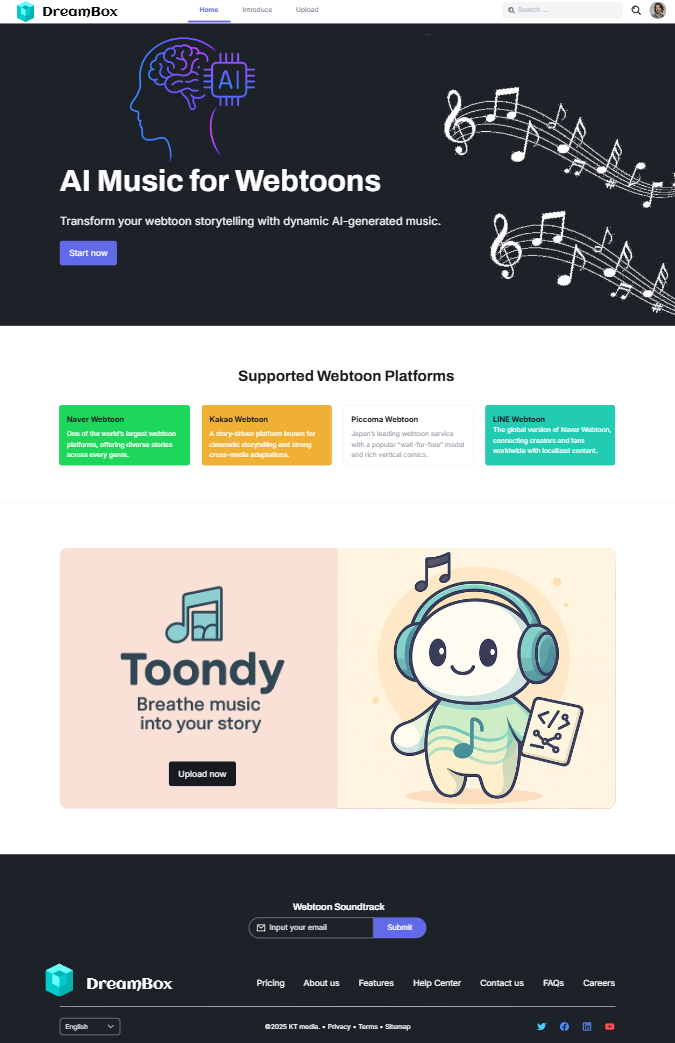
생성된 키워드 기반으로 **Meta의 MusicGen 모델**을 활용하여 6개 세그먼트의 음악 시퀀스를 생성합니다.  
각 세그먼트는 콘텐츠 전개 흐름에 따라 자동 분할되며, 감정 변화에 따라 구간별로 음악이 동기화됩니다.

**5. 오디오 렌더링 및 저장**

추출된 시퀀스는 wav 포맷으로 렌더링 및 저장되며,  
각 구간은 관련 **Metadata (keywords, mood, genre, duration 등)** 와 함께 저장되어 이후 감상 중 사용자 설정 기반으로 커스터마이징이 가능하게 구성됩니다.

# UX 설계

[Visily AI](https://app.visily.ai/) 활용한 Toondy의 UX 설계 입니다.



# 데이터 설계

본 시스템은 웹툰 이미지 또는 소설 텍스트를 입력으로 받아, 감정 및 키워드를 분석한 후 해당 분위기에 어울리는 음악을 생성하는 구조로 설계되었다. 이를 위해 텍스트, 이미지, 분석 결과, 생성 결과 등의 다양한 데이터를 다루며, 단계별로 적절한 데이터 구조를 활용합니다.

먼저 콘텐츠 입력 단계에서는 웹툰의 경우 웹페이지로부터 이미지 데이터를 크롤링하며 각 이미지는 파일 경로 리스트로 관리됩니다. 이때 pytesseract를 이용하여 OCR을 수행하고 추출된 텍스트는 대사 단위의 리스트로 저장됩니다. 반면 소설의 경우 .txt 파일을 입력으로 받아 전체 본문을 읽은 뒤, 챕터 단위로 분리된 리스트(chapter\_data)와 전체 단어 수 등의 메타데이터를 함께 저장합니다.

감정 분석 단계에서는 텍스트로부터 KoNLPy를 이용한 형태소 분석과 KeyBERT 기반 키워드 추출이 수행되며, 이미지에서는 GPT-4o Vision API를 통해 감정 및 분위기를 추출합니다. 이 두 소스를 통합하여 최종 키워드 리스트를 구성하고, 등장 빈도와 키워드 사전에 기반해 genre, mood, era, music\_style 값을 분류합니다.  
이후 프롬프트 생성 단계에서는 위의 분석 결과를 기반으로 facebook/musicgen-small 모델이 사용할 수 있는 자연어 입력 프롬프트를 자동 생성하며, 음악 생성 결과는 .wav 형식의 오디오 파일로 저장됩니다. 이와 함께 프롬프트와 분석 정보는 .txt, 키워드는 wordcloud 형태의 .png 이미지로 시각화되어 함께 저장됩니다.

모든 데이터는 콘텐츠 단위로 output 디렉토리에 폴더별로 구조화되어 저장되며, 반복 실행 시 캐시를 활용하여 분석 속도를 최적화할 수 있도록 설계되었습니다.

# 프로그램 모듈 구조, 모델 및 알고리즘 설계

**1. 프로그램 모듈 구조**

* [main.p](http://main.py)y  
  : 전체 파이프라인 제어, 사용자 입력 인자 해석,   
  콘텐츠 유형(webtoon/novel) 분기 처리
* webtoon\_processor.py  
  : 웹툰 이미지 크롤링 (BeautifulSoup), OCR(pytesseract) 수행, 텍스트/이미지 경로 반환
* novel\_processor.py  
  : .txt 소설 파일 읽기, 챕터 단위 분리, 단어 수 및 메타데이터 생성
* keyword\_extractor.py  
  : 형태소 분석 (KoNLPy), 키워드 추출 (KeyBERT), 감정 분석 (GPT-4o Vision API), 장르/분위기/시대/스타일 분류
* music\_generator.py  
  : 분석 결과 기반 프롬프트 생성, MusicGen 모델로 6세그먼트 음악 생성,   
  pydub로 crossfade 합성
* [utils.py](http://utils.py)  
  : 생성 결과 저장 (.wav, .txt), 키워드 워드클라우드 시각화, 디렉터리 구조 및 캐시 관리

**2. 알고리즘 구조**

(1) 키워드 추출

○ 텍스트 기반 키워드 추출

1. 형태소 분석  
 한국어 텍스트의 경우 KoNLPy 라이브러리의 Okt 분석기를 사용하여   
 명사 및 핵심 단어를 분리

2. 키워드 추출  
 전처리된 텍스트에서 KeyBERT를 활용하여 문맥 기반 핵심 키워드를 추출.   
 NLTK는 영어 콘텐츠에 한해 불용어(stopword) 제거 및 토큰화에 활용

○ 이미지 기반 감정 키워드 추출

1. 웹툰의 합성 이미지(combined\_image\_path)를 pytesseract를 통해 OCR 수행

2. 텍스트 외에도 OpenAI GPT-4o Vision API를 이용하여 이미지에서 감정적 분위기,  
 장면 정보, 배경 요소 등을 추출

○ 텍스트 + 이미지 키워드 결합

1. OCR 텍스트 기반 키워드와 Vision 기반 감정 키워드를   
 deduplication(중복 제거) 및 빈도/우선도 기반 weighting 후 통합

(2) 분류 (Classification)

1. 장르(Genre), 분위기(Mood), 시대(Era), 음악 스타일(Music Style)은   
 사전 정의된 키워드

2. 세트(\*\_keywords)에 기반한 룰 기반 분류 알고리즘으로 결정

(3) 음악 생성 (Music Generation)

1. HuggingFace 모델인 facebook/musicgen-small을 사용하여 프롬프트 기반   
 음악을 생성

2. 세그먼트는 pydub을 활용해 crossfade 처리 후 3분짜리 .wav로 최종 합성

(4) 결과 저장 및 시각화 (Saving & Visualization)

1. .wav: 최종 생성된 음악 파일

2. .txt: 생성에 사용된 키워드, 장르, 분위기, 생성 프롬프트 요약

3. .png: wordcloud 라이브러리를 사용하여 키워드 시각화

# 테스트 계획

1. 테스트 목표

본 테스트는 콘텐츠 기반 감정 분석 및 음악 생성 시스템의 기능 정확성, 처리 안정성, 결과 일관성을 검증하는 것을 목표로 합니다.

웹툰 및 소설이라는 두 입력 유형에 대해 파이프라인이 정확히 작동하는지를 확인하며, 생성된 결과물이 입력과 논리적으로 연결되는지를 테스트합니다.

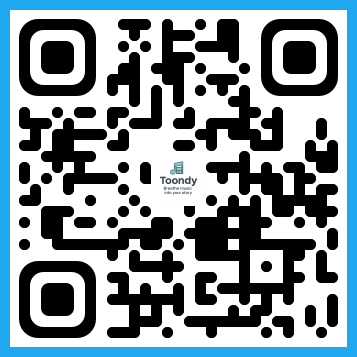
2. 테스트 항목

1. 콘텐츠 입력 처리 :   
   웹툰 URL 또는 소설 파일 입력 시 적절히 분기되고 정상적으로 로드되는지 확인
2. 이미지 크롤링 및 OCR :   
   웹툰 이미지가 잘 수집되고 텍스트가 추출되는지 확인
3. 텍스트 전처리 및 키워드 추출 :  
   KeyBERT 기반 핵심 키워드가 추출되고 KoNLPy 분석이 정상적으로 수행되는지 확인
4. 이미지 감정 분석 :  
   GPT-4o Vision API를 통해 이미지에서 감정 키워드가 잘 생성되는지 확인
5. 키워드 결합 및 분류 :  
   텍스트+이미지 키워드가 중복 없이 병합되고 장르/분위기/시대/음악스타일이 분류되는지 확인
6. 음악 생성 (MusicGen) :  
   프롬프트 기반 음악 생성이 자연스럽게 작동하는지, 6세그먼트가 잘 이어지는지 확인
7. 결과 저장 및 시각화 :  
   .wav, .txt, .png 파일이 제대로 저장되는지, 디렉토리 구조가 유지되는지 확인
8. 에러 및 예외 처리 :  
   입력 누락, 이미지 없음, API 응답 실패 등 예외 상황 대응 확인

# 구현

다음의 QR코드를 스캔하여 Hugging Face에 배포된 Toondy 를 직접 사용해 볼 수 있습니다.

BGM을 만들고 싶은 웹툰의 이미지나 웹소설의 txt 파일을 넣고, openAI 키를 입력해야 합니다.



다음은 시연 영상이 올라간 youtube 영상 링크입니다.

<https://youtu.be/iDXE8Kmgd40>

# 결과

본 프로젝트는 웹툰 및 웹소설과 같은 텍스트 기반 콘텐츠의 감정 흐름을 실시간으로 분석하고, 이에 적절한 BGM을 자동 생성·동기화함으로써 **사용자의 몰입감을 실질적으로 향상시키는 감상 환경을 구현**하는 데 중점을 두었습니다.  
 감정 전개에 맞춰 생성된 배경음악은 사용자의 감정 몰입도를 증진시키고, **작품의 스토리라인에 대한 공감 수치를 높이는 데 기여**하였습니다.

향후 고도화를 위한 과제로는, 보다 섬세하고 풍부한 감정 인식을 가능하게 하기 위해 **감정 상태의 벡터화** 기법을 도입하고자 합니다. 이를 통해 단순한 감정 분류를 넘어 **복합적 감정 상태까지 반영 가능한 음악 생성이 가능해질 것**으로 기대됩니다.   
또한 이미지 기반 콘텐츠 분석 과정에서 **불필요한 토큰 낭비를 줄이고 분석 효율을 향상시키기 위해**, Grayscale 변환을 활용한 **자체 이미지 분석 모듈의 개발도 계획 중**에 있습니다.

향후에는 주요 웹툰/웹소설 플랫폼과의 협업을 통해, **사용자가 콘텐츠 감상 중 실시간으로 감정에 동기화된 음악을 경험할 수 있는 통합 감상 환경**을 구축하고자 합니다. 이를 통해 독자에게 한층 더 몰입도 높은 감정적 경험을 제공하며, **스토리 기반 AI 음악 동기화 기술의 상용화 가능성**을 본격적으로 검증할 계획입니다.

# 감사의 글

본 과제 수행에 있어 크고 작은 도움을 아낌없이 주신 모든 분들께 깊은 감사를 드립니다.

먼저, 생성형 AI의 활용 가능성과 실제 구현에 대한 통찰을 제공해준 GPT 기술과 오픈소스 커뮤니티에 감사드립니다. 실제 프로젝트와 연결된 실습 환경을 제공해주신 패스트캠퍼스와, 이 기회를 가능하게 해주신 KT에도 진심으로 감사드립니다. 끊임없는 피드백과 실전 중심의 멘토링으로 방향을 잡아주신 강태욱 강사님, 프로젝트 전 과정에서 든든한 다리 역할을 해주신 조교님, 그리고 수강생들의 흐름이 끊기지 않도록 실무 지원을 도와주신 패스트캠퍼스 운영팀 여러분께도 깊은 감사를 전합니다.

끝으로, 이 과제를 통해 기술의 의미와 사람 간의 연결을 다시 생각하게 된 이 시간에 감사드리며, 앞으로도 더 깊이 있는 연구와 구현으로 이 마음을 이어가겠습니다.

# 레퍼런스

1. https://github.com/mac999/LLM-RAG-Agent-Tutorial
2. https://github.com/MaartenGr/KeyBERT
3. <http://konlpy.org>
4. <https://github.com/madmaze/pytesseract>
5. <https://www.crummy.com/software/BeautifulSoup/>
6. <https://huggingface.co/facebook/musicgen-small>
7. <https://github.com/jiaaro/pydub>
8. <https://github.com/amueller/word_cloud>