

<b>PJT명</b>	API 요청을 통한 데이터를 직접 수집하고 생성형 AI를 이용한 데이터 분류, 요약된 내용을 오디오 파일로 변환하기	
<b>단계</b>	[Python PJT]	
<b>진행일자</b>	2025.07.25	
<b>예상 구현 시간</b>	필수기능	5H
	추가기능	2H
	심화기능	1H

## 1. 목표

- Python 기초 문법과 라이브러리 활용 능력을 습득한다.
- 데이터 수집 및 분석 프로세스에 대하여 이해한다.
- gTTS를 이용해 텍스트를 음성으로 변환해본다.
- 생성형 AI를 활용하여 데이터를 분류해본다.
- 프로젝트를 단계적으로 확장해 나가고, 연결하는 작업을 경험한다.

## 2. 준비사항

### 1) 프로젝트 구조

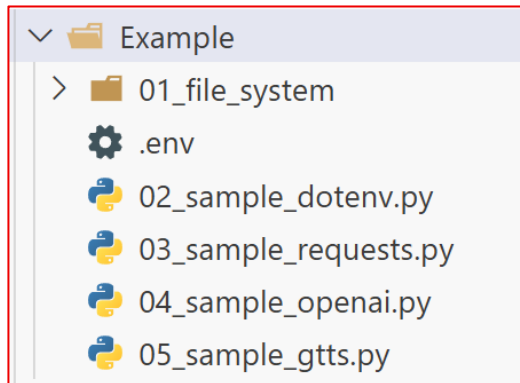
- 프로젝트는 총 하나의 폴더, 9개의 Python 코드, 그 외 2가지 파일로 구성되어 있음
  1. problem\_A1.py ~ problem\_C3.py: 요구사항 구현을 위한 스켈레톤 코드
    - problem\_XN.py의 형태로 이름이 지어짐
    - X: 과제 그룹
    - N: 요구사항 번호
  2. Example 폴더: 아래 2) 사용 데이터에 설명된 Example 폴더
  3. .env 파일: API 키를 관리하기 위한 파일

4. requirements.txt 파일: 가상환경 생성을 위한 requirements.txt 파일

## 2) 사용 데이터

- Example 폴더: 이번 프로젝트 해결을 위해 알아야 하는, 혹은 직접적인 도움이 될 수 있는 코드

예시:



```
# .env 파일 로드
load_dotenv()

# 환경 변수 접근
API_KEY = os.getenv('API_KEY')
MY_DATABASE_URL = os.getenv('DATABASE_URL')
DEBUG_MODE = os.getenv('DEBUG_MODE', 'False')
```

## 3) 개발언어 및 툴

- Python 3.11+ / Visual Studio Code

## 4) 필수 라이브러리 / 오픈소스

- Python 내장 json 모듈
- Python 내장 pathlib 모듈
- Python dotenv 패키지
- Python requests 패키지
- Python gTTS 패키지
- Python openai 패키지

### 3.작업 순서

- 1) 팀원과 같이 요구사항(기본/추가/심화)을 확인하고, GitLab에 프로젝트를 생성한다.
  - 프로젝트 이름은 02-pjt로 지정한다.
  - 각 반 담당 강사님을 Maintainer로 설정한다.
- 2) 제공된 Example의 코드를 확인하고, 요구사항에 필요한 코드를 파악한다.
- 3) 팀원과 합의하여 과제 그룹을 선정한다.
  - 과제 그룹 별로 요구사항이 다르며, 난이도가 상이하다.
  - 팀원과 합의하여 적정 수준의 그룹을 선정하도록 한다.
- 4) 선택한 과제 그룹의 필수 요구사항을 구현한다.
- 5) 작성한 코드들을 정리하고, README를 작성한다.
- 6) README 작성이 완료되면 심화 학습을 진행한다.
- 7) 제출 기한에 맞춰 모든 산출물이 GitLab에 업로드 될 수 있도록 한다.

## 4. 요구사항

AI 기반 도서 분석과 창작 지원 통합 솔루션 서비스를 구축하려고 한다. 다양한 장르와 형태의 도서 데이터를 수집 및 관리하고, 이를 기반으로 AI 기반 도서 분석 기능을 설계 및 구현한다. 또한 작가들에게 아이디어 발상 지원, 플롯 구성 제안, 문장 교정 및 스타일 개선 등 창작 전반에 걸친 다양한 AI 기반 창작 지원 기능을 제공한다. 또한 도서에 대한 사용자 리뷰 및 감상평 공유 커뮤니티 기능을 제공하여, 사용자들이 활발하게 소통하고 정보를 교환할 수 있는 기능을 제공한다. 사용자는 자신이 읽은 도서를 평가하고, 다른 사용자의 리뷰를 참고하여 다음 도서를 선택하는데 도움을 받을 수 있다. 나아가, 관심 도서 목록을 맞춤형으로 구성하는 등 다양한 편의 기능을 제공한다. 팀원과 상의하여 아래 요구사항을 만족할 수 있도록 요구 사항 명세서를 작성 및 구현해보자.

본격적인 서비스 구현에 앞서, API를 활용하여 데이터를 받아오고, 필요한 형태로 정리하는 연습을 진행해보자.

과제 그룹 A, B, C 중 하나를 선택해서 진행한다.

● 요구사항 예시(참고용)

- 아래의 내용을 참고하여 추가적인 아이디어에 대해 요구사항을 추가 또는 수정하여 기능을 구현한다. 단, **필수 기능은 구현해야 하며, 수정할 수 없다.**

번호	분류	요구사항명	요구사항 상세	우선순위
기능적 요구사항				
F01	도서 데이터	민감 정보 관리	API Key와 같은 민감 데이터를 코드에서 분리하여 관리하는 기능	필수
F02	도서 데이터	API 호출	API를 활용해 제목을 기준으로 도서를 검색하는 기능	필수
F03	도서 데이터	데이터 변환	응답 데이터를 JSON 형식으로 해석하는 기능	필수
F04	도서 데이터	데이터 변환	응답 데이터를 필요한 형태로 변환하는 기능	필수
F05	도서 데이터	데이터 저장	수집한 데이터를 형식을 갖춰 파일의 형태로 저장하는 기능	필수
F06	도서 데이터	음성 파일 생성	gTTS를 이용해 제목을 음성 파일로 변환하고 저장하는 기능	필수
F07	도서 데이터	API 호출	API를 활용해 주제를 기준으로 도서를 검색하는 기능	필수
F08	도서 데이터	데이터 집계	응답 데이터를 바탕으로 정보에 대한 통계를 구하는 기능	필수
F09	도서 데이터	데이터 정리	응답 데이터를 일정한 기준으로 분류하여 정리하는 기능	필수
F10	도서 데이터	음성 파일 생성	여러 데이터를 순차적으로 음성 파일로 변환하여 저장하는 기능	필수
F11	도서 데이터	API 호출	API에서 한 번에 검색을 허용하는 범위를 넘어서는 데이터를 검색하는 기능	필수
F12	도서 데이터	데이터 정리	응답 데이터를 특정 기준으로 정리하는 기능	필수

F13	도서 데이터	데이터 저장	수집한 데이터를 형식을 갖춰 파일의 형태로 저장하는 기능	필수
F14	도서 데이터	AI 기반 데이터 분류	생성형 AI를 활용하여 도서를 분류하는 기능	필수
F15	도서 데이터	데이터 저장	정리한 데이터를 형식을 갖춰 파일의 형태로 저장하는 기능	필수
F16	도서 데이터	데이터 수집	응답 데이터에서 필요한 정보를 추출하여 저장하는 기능	필수
F17	도서 데이터	음성 파일 생성	gTTS를 이용해 저장된 데이터를 음성 파일로 변환하여 저장하는 기능	필수
F18	도서 데이터	AI 기반 요약 정리	생성형 AI를 활용해 결과를 요약 정리하는 기능	심화
F19	도서 데이터	음성 파일 생성	요약된 결과를 음성 파일로 변환하는 기능	심화
F20	도서 데이터	챗봇 구현	생성형 AI를 활용해 존재하는 기능을 챗봇 UI에 적용하는 기능	심화
F21	도서 데이터	API 활용	생성형 AI를 활용하여 다른 도서 데이터 API를 활용하는 기능	심화
...	...	...	...	...

## 1) 기본(필수) 기능 (그룹 A)

### ① API 요청 함수 작성하기

도서 정보를 검색하고 출력하는 `fetch_books` 함수를 완성한다.

- API 키는 ".env" 파일에 저장하여 `dotenv` 라이브러리를 통해 불러온다.
  - `requests` 라이브러리를 활용하여 검색 요청을 보낸다.
  - 응답 데이터를 JSON 형식으로 변환한 후 `fetch_books` 함수의 실행 결과로 반환한다.
  - 완성된 `fetch_books` 함수를 사용해, 펭귄 관련 도서 10권을 검색하고, 번호를 붙여 제목을 출력한다.
- 요구사항 번호: F01, F02, F03, F04
  - 주어진 `problem_A1.py`을 수정하여 구현
  - 구현해야 할 화면 (콘솔 출력)

1. 핑크펭귄 - 완전히 새롭고 전혀 다른 그 무언가!, 개정판
2. 펭귄 365
3. 너에게 들려주는 단단한 말
4. 나를 소모하지 않는 현명한 태도에 관하여
5. 최재천의 동물대탐험 5 : 황제펭귄의 행진 - 남극의 육아 천재, 황제펭귄
6. 살아갈 날들을 위한 괴테의 시
7. 매거진 B (Magazine B) Vol.10 : 펭귄박스 (PENGUIN) - 국문판 2012.10
8. 펭귄은 너무해
9. 펭귄 호텔
10. 펭귄뉴스

실시간 데이터이기에 출력 결과가 상이할 수 있음

## ② 도서 데이터 가공 및 저장

특정 키워드 ("컴퓨터")로 검색된 도서 정보를 저장하고 필요한 필드만 정리하여 파일로 저장하는 `save_books_to_file` 함수를 완성한다.

1. 앞서 구현한 `fetch_books` 함수의 코드를 활용한다.
2. 파일 저장을 위한 "output" 폴더를 생성한다.
  - 이미 폴더가 존재해도 에러가 발생하지 않아야 한다
3. 검색 결과 데이터를 받아와 지정한 파일에 저장한다.
  - "제목", "저자", "출판사", "가격" 항목을  
마크다운 형식으로 출력하고 파일로 저장한다.
  - 가격은 정가(`priceStandard`)를 활용한다.
4. 데이터는 `output/computer_books.txt` 파일로 저장한다.

- 요구사항 번호: F05
- 주어진 `problem_A2.py`을 수정하여 구현
- 구현해야 할 화면 (콘솔 출력, 결과 일부 생략)

```
## 제목: 파인만의 컴퓨터 강의 - 양자 컴퓨터를 고안한 천재가 들려주는
* 저자: 리처드 파인만 (지은이), 토니 헤이 (역은이), 서환수 (옮긴이)
* 출판사: 한빛미디어, * 가격: 29000원
-----
## 제목: 양자컴퓨터의 미래, 불확정성,
* 저자: 데이비드 린들리 (지은이), 박배식 (옮긴이)
* 출판사: 시스템아, * 가격: 25000원
-----
## 제목: 혼자 공부하는 컴퓨터 구조 + 운영체제 - 1:1 과외하듯 배우는
* 저자: 강민철 (지은이)
* 출판사: 한빛미디어, * 가격: 28000원
-----
```

실시간 데이터이기에 출력 결과가 상이할 수 있음



### ③ TTS 파일 생성하기



gTTS 라이브러리를 이용해 검색된 도서의 첫 번째 제목을 음성 파일로 저장하는 `create_tts_file` 함수를 완성한다.

1. 먼저 주어진 키워드를 사용하여 알라딘 API에서 도서를 검색하고 제목을 반환하는 `fetch_book_title` 함수를 완성한다.
2. `create_tts_file` 함수에서 완성한 `fetch_book_title`을 호출하여 제목을 가져오고, 해당 제목을 음성 변환(TTS)하여 오디오 파일로 저장한다.
3. 결과는 `output/book_title.mp3` 파일로 저장한다.

- 요구사항 번호: F06
- 주어진 `problem_A3.py`을 수정하여 구현
- 구현해야 할 화면 (콘솔 출력)

음성 파일이 `output/book_title.mp3`로 저장되었습니다.

- 생성 파일

▽  output  
     book\_title.mp3

## 2) 기본(필수) 기능 (그룹 B)

### ① 다양한 주제 데이터 수집 및 통합하기

"자연", "예술", "기술" 세 가지 주제에 대해 각각 30개의 도서를 조회하고 평균 가격과 평균 출판 연도를 출력하는 `collect_and_analyze_books` 함수를 완성한다.

1. 먼저 주제별로 도서 정보를 조회하여 수집하는 `fetch_books_by_topic` 함수를 완성한다.
2. `collect_and_analyze_books` 내부에 정의된 `topic_stats` 딕셔너리에 각 주제별 통계(평균 가격, 평균 출판 연도)를 계산하여 저장한다.
  - 가격은 정가(`priceStandard`)를 활용한다.
  - 출판 연도는 `pubDate`의 데이터에서 추출하여 활용한다.
3. 결과를 화면에 출력한다.

- 요구사항 번호: F07, F08
- 주어진 `problem_B1.py`을 수정하여 구현
- 구현해야 할 화면 (콘솔 출력)

통합된 도서 수: 90

전체 통계: 평균 가격 22487원, 평균 출판 연도 2020

자연 주제 통계:

평균 가격: 22630.00원

평균 출판 연도: 2019

예술 주제 통계:

평균 가격: 24430.00원

평균 출판 연도: 2019

기술 주제 통계:

평균 가격: 20400.00원

평균 출판 연도: 2021

실시간 데이터이기에 출력 결과가 상이할 수 있음

## ② 데이터 분류 및 저장하기

“경제” 관련 도서 데이터를 50개 수집하고, 할인 중인 책과 정가인 책으로 나누어 JSON 형식으로 저장하는 `classify_and_save_books` 함수를 완성한다.

1. 앞서 구현한 `fetch_books_by_topic` 함수의 코드를 활용한다.
2. 할인 가격과 정가를 비교하여 할인 여부를 판단한다.
  - 할인 가격은 `priceSales`, 정가는 `priceStandard`를 활용한다.
3. 할인 중인 책 데이터는 `output/discounted_books.json`, 정가인 책 데이터는 `output/regular_priced_books.json`로 저장한다.
4. 할인 중인 책의 개수와 정가인 책의 개수를 출력한다.

- 요구사항 번호: F09
- 주어진 `problem_B2.py`을 수정하여 구현
- 구현해야 할 화면 (콘솔 출력)

할인 중인 책 44개, 정가인 책 6개를 저장했습니다.

- 생성 파일

```
✓ output
  {} discounted_books.json
  {} regular_priced_books.json
```

- output/discounted\_books.json 예시 (일부)

```
[
  {
    "title": "세계 경제 지각 변동 -",
    "link": "https://www.aladin.co.",
    "author": "박종훈 (지은이)",
    "pubDate": "2025-06-27",
    "description": "지금까지 경험해",
    "isbn": "K282030989",
    "isbn13": "9791198987631",
    "itemId": 366397343,
    "priceSales": 19800,
    "priceStandard": 22000,
    "mallType": "BOOK",
```

- output/regular\_priced\_books.json 예시 (일부)

```
[
  {
    "title": "경제금융용어 700선",
    "link": "https://www.aladin.co.",
    "author": "한국은행 (엮은이)",
    "pubDate": "2020-08-18",
    "description": "",
    "isbn": "K012632571",
    "isbn13": "9791155383933",
    "itemId": 249222069,
    "priceSales": 8000,
    "priceStandard": 8000,
    "mallType": "BOOK",
```

실시간 데이터이기에 출력 결과가 상이할 수 있음

### ③ TTS를 활용하여 데이터 요약 오디오 만들기

한국문학 주제를 검색하여 description이 있는 5개의 도서 제목과 요약을 JSON으로 정리하고, 제목과 요약에 대하여 오디오 파일을 만드는 process\_korean\_literature\_books 함수를 완성한다.

1. 앞서 구현한 fetch\_books\_by\_topic 함수의 코드를 활용한다.
2. 도서에 요약(description) 정보가 있을 경우에만 리스트로 정리하여 저장하는 save\_books\_to\_json 함수를 작성한다.
  - 최대 5개의 정보를 저장한다.
  - output/korean\_literature\_books.json에 파일로 저장한다.
3. save\_books\_to\_json으로 만들어진 JSON 파일을 읽어 한국어 음성으로 변환하는 create\_tts\_file 함수를 작성한다.
  - 각 도서의 제목과 요약을 결합하여 생성한다.
  - n번째 JSON 파일의 내용을 output/book\_n.mp3 파일로 저장한다.
4. 몇 개의 책 정보를 처리했는지 출력한다.

- 요구사항 번호: F10
- 주어진 problem\_B3.py을 수정하여 구현
- 구현해야 할 화면 (콘솔 출력)

5개의 책 정보를 처리하고 음성 파일로 저장했습니다.

- 생성 파일

✓ 📁 output

- 🔊 book\_1.mp3
- 🔊 book\_2.mp3
- 🔊 book\_3.mp3
- 🔊 book\_4.mp3
- 🔊 book\_5.mp3
- { } korean\_literature\_books.json

- output/korean\_literature\_books.json 예시 (일부)

```
[  
  {  
    "제목": "EBS 수능완성 국어영역 독서·문학",  
    "요약": "최신 수능 출제 경향과 유형을 분석한",  
  },  
  {  
    "제목": "EBS 수능완성 국어영역 독서·문학",  
    "요약": "최신 수능 출제 경향과 유형을 분석한",  
  },  
]
```

실시간 데이터이기에 출력 결과가 상이할 수 있음

### 3) 기본(필수) 기능 (그룹 C)

#### ① 데이터 검색 기능 확장 및 분석하기

“인공지능” 관련 도서 100개를 알라딘 API에서 검색하고, 가격이 높은 순서대로 상위 10개 도서를 JSON 파일로 저장하는 `process_ai_books` 함수를 완성한다.

1. 알라딘 API는 한 번에 50개의 데이터만 응답해준다. 문서를 참고하여 중복되지 않은 100개의 책의 데이터를 수집하는 `fetch_books_by_topic` 함수를 작성한다.
2. 100개의 도서 데이터 중 가격이 높은 순서대로 상위 10개의 도서 정보를 출력한다.
  - 가격은 정가(`priceStandard`)를 활용한다.
3. 제목과 저자 정보를 요약하여 `output/ai_top10_books.json` 파일로 저장한다.

- 요구사항 번호: F11, F12, F13
- 주어진 `problem_C1.py`을 수정하여 구현
- 구현해야 할 화면 (콘솔 출력)

가격이 높은 순서대로 상위 10개 도서:

1. 제목: the AI GRAPHICS : 인공지능, 캐릭터, 웹툰, 패션, 세계관,
  2. 제목: 인공지능 1 - 현대적 접근방식, 제4판, 가격: 42000원
  3. 제목: AI 텐베거 투자 - 뉴사이클에 진입한 AI 혁명 산업, 10배 증
  4. 제목: 인공지능 2 - 현대적 접근방식, 제4판, 가격: 38000원
  5. 제목: 제미나이 인공지능 프로그래밍 - 라마인덱스, 랭체인과 함께
  6. 제목: 인공지능 - 파이썬으로 배우는 머신러닝과 딥러닝, 2판, 가
  7. 제목: 인공지능 수학 with 파이썬 - 수학자와 함께하는, 가격: 32
  8. 제목: 파이썬으로 만드는 인공지능, 가격: 31000원
  9. 제목: 모두의 인공지능 with 파이썬 - 머신러닝, 딥러닝부터 생성
  10. 제목: 인공지능 시대의 선형대수학 - 개정3판, 가격: 30000원
- 'ai\_top10\_books.json' 파일이 생성되었습니다.

- 생성 파일

▼  output  
|  ai\_top10\_books.json

- output/ai\_top10\_books.json 예시 (일부)

```
[
  {
    "제목": "the AI GRAPHICS : 인공지능, 캐릭터",
    "저자": "김성완, 김한재, 류내원, 박상준, 이",
  },
  {
    "제목": "인공지능 1 - 현대적 접근방식, 제4판",
    "저자": "스튜어트 러셀, 피터 노빅 (지은이),",
  },
]
```

실시간 데이터이기에 출력 결과가 상이할 수 있음



## ② 생성형 AI를 활용한 키워드 기반 도서 분류하기

검색한 도서를 생성형 AI를 활용하여 분류하고 파일로 저장하는 `process_reading_books` 함수를 완성한다.

1. 앞서 구현한 `fetch_books_by_topic` 함수의 코드를 활용하여, "독서"와 관련된 도서 100개를 수집한다.
2. 검색 결과를 바탕으로 생성형 AI를 활용하여 제목에 "습관", "시간관리", "독서법", "기타"로 각 도서를 분류한다.
  - 이 때, 생성형 AI의 응답이 JSON으로 반환되도록 프롬프트 및 파라미터(`temperature`, `top_p`)를 수정한다.
3. 분류된 데이터를 `output/reading_habits.json` 파일로 저장한다.

- 요구사항 번호: F14, F15
- 주어진 `problem_C2.py`을 수정하여 구현
- 구현해야 할 화면 (콘솔 출력)

'reading\_habits.json' 파일이 생성되었습니다.

- 생성 파일

output  
reading\_habits.json

- `output/reading_habits.json` 예시 (일부)

```
{  
  "습관": [  
    "독서평설 첫걸음 2025.7 - 만 5  
    "독서평설 첫걸음 2025.8 - 만 5  
    "독서평설 첫걸음 2025.6 - 만 5  
    "뇌가 젊어지는 독서 습관 귀독서  
    "독서는 절대 나를 배신하지 않는다  
  ],  
  "시간 관리": [  
    "EBS 하루 3개 1등급 국어독서 2
```

실시간 데이터이기에 출력 결과가 상이할 수 있음

### ③ TTS를 활용하여 주제별 오디오 파일 생성하기

“음악” 관련 도서 10개를 검색하여 각 도서의 제목, 저자, 그리고 간단한 소개를 포함한 텍스트를 저장하고, 이를 기반으로 오디오 파일을 생성하는 `process_music_books` 함수를 완성한다.

1. 알라딘 API를 활용하여 “음악” 관련 도서를 10개 검색한다.
2. 각 도서의 제목, 저자, 그리고 책 소개를 추출하여 `output/music_books_info.txt` 텍스트 파일로 저장한다.
3. 저장된 텍스트를 읽어 gTTS 라이브러리로 음성 변환(TTS)하여 `output/music_books.mp3` 파일로 저장한다.

- 요구사항 번호: F16, F17
- 주어진 `problem_C3.py`을 수정하여 구현
- 구현해야 할 화면 (콘솔 출력)

output/music\_books\_info.txt 파일이 생성되었습니다.  
output/music\_books.mp3 파일이 생성되었습니다.  
모든 작업이 완료되었습니다.

- 생성 파일

output  
music\_books\_info.txt  
music\_books.mp3

- `output/music_books_info.txt` 예시 (일부)

제목: 전쟁과 음악 - 양차 대전과 냉전  
제목: 음악소설집 音樂小說集, 저자: 김  
제목: 음악의 역사 - 소리로 말하고 함  
제목: 턴! 턴! 턴! Turn! Turn! Turn!  
제목: 캔디팡팡 음악종합장 4단, 저자:

실시간 데이터이기에 출력 결과가 상이할 수 있음

#### 4) 심화 학습

프롬프트를 진행하면서 다음과 같은 심화 학습을 시도해볼 수 있다.

**① 도서 데이터를 생성형 AI를 활용하여 분석 및 요약하고, 요약된 데이터를 오디오로 생성해보기**

그룹별로 생성된 데이터를 생성형 AI의 API를 활용해 요약 정리한 다음, 결과를 음성으로 받아본다.

- 요구사항 번호: F18, F19

**② 챗봇 동작을 할 수 있도록 전체 코드를 정리해보기**

구현한 코드를 챗봇 UI를 이용해 사용할 수 있도록 생성형 AI를 활용해 개발해본다.

- 요구사항 번호: F20

**③ 다른 도서 API를 확인하고 프로젝트 내에서 활용할 방법을 연구해보기**

생성형 AI를 활용해 도서 정보를 제공하는 다른 API를 사용해보고, 같은 프로젝트 내에서 활용법을 고민 및 개발해본다.

- 요구사항 번호: F21

- API 목록

1. 국가 자료 종합 목록

<https://www.nl.go.kr/NL/contents/N31101030400.do>

2. 네이버 도서검색 API

<https://developers.naver.com/docs/serviceapi/search/book/book.md>

3. 카카오 도서검색 API

<https://developers.kakao.com/docs/latest/ko/daum-search/dev-guide>

상기된 과제를 전부 완료한 경우 명시된 요구사항 외 추가 개발을 자유롭게 진행한다.

## 5. 참고자료

- requests  
<https://requests.readthedocs.io/en/latest/>
- gTTS  
<https://gtts.readthedocs.io/en/latest/>
- python-dotenv  
<https://pypi.org/project/python-dotenv/>
- Python OpenAI Library  
<https://platform.openai.com/docs/libraries?language=python>  
<https://github.com/openai/openai-python>

## 6. 결과

제출 기한은 진행일 18시까지이므로 제출 기한을 지킬 수 있도록 한다. 제출은 GitLab을 통해서 이뤄진다.

- 산출물과 제출
  - 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 상세히 기록한 README.md
  - 완성된 각 문제 별 소스코드 및 실행 화면 캡처본
  - 프로젝트 이름은 02\_pjt로 지정, 각자의 계정에 생성할 것
  - 각 반 담당 강사님을 Maintainer로 설정

- 끝 -