성결대학교
SUNGKYUL UNIVERSITY

소프트웨어공학
1팀

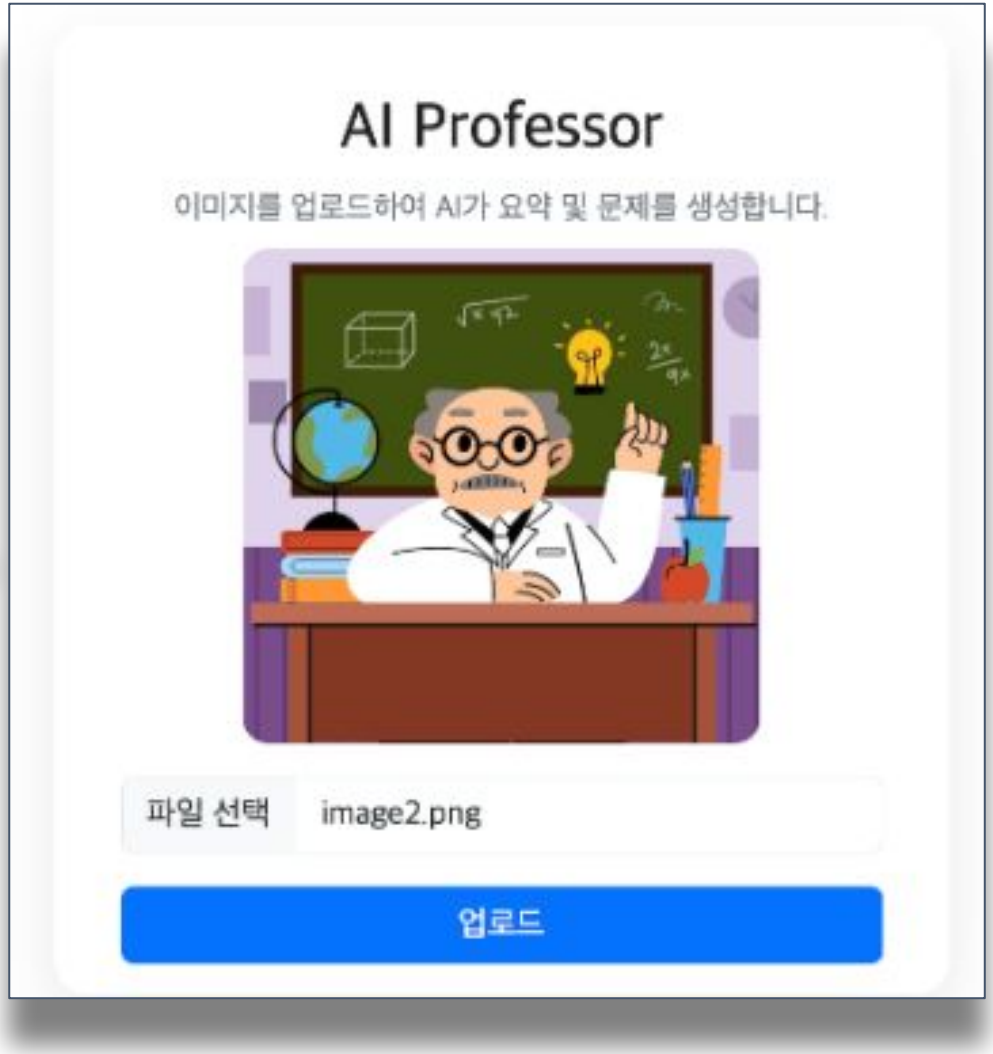
- 류진서 (PPT 제작 및 시연 영상 제작)
- 서예원 (GITHUB 관리 및 코드 총괄)
- 김재현 (OCR 담당)
- 김민우 (VLM 담당)
- 황예림 (UI/UX 담당)
- 김두현 (LLM 담당)
- 오기주 (LLM 담당)

주제 논의

- AI 교수(* 선택)
- 영수증 및 명함 정리
- AI 가짜 뉴스 판별기
- AI 여행 사진 정리 및 스토리텔링
- AI 일정표 요약 및 리마인더
- 냉장고 재료 요리 추천
- 코딩 문제 해설기

AI PROFESSOR

수업 자료 요약



퀴즈 문제 생성

AI PROFESSOR - 흐름

1. 이미지 촬영 및 PDF 파일 준비.

2. 업로드.

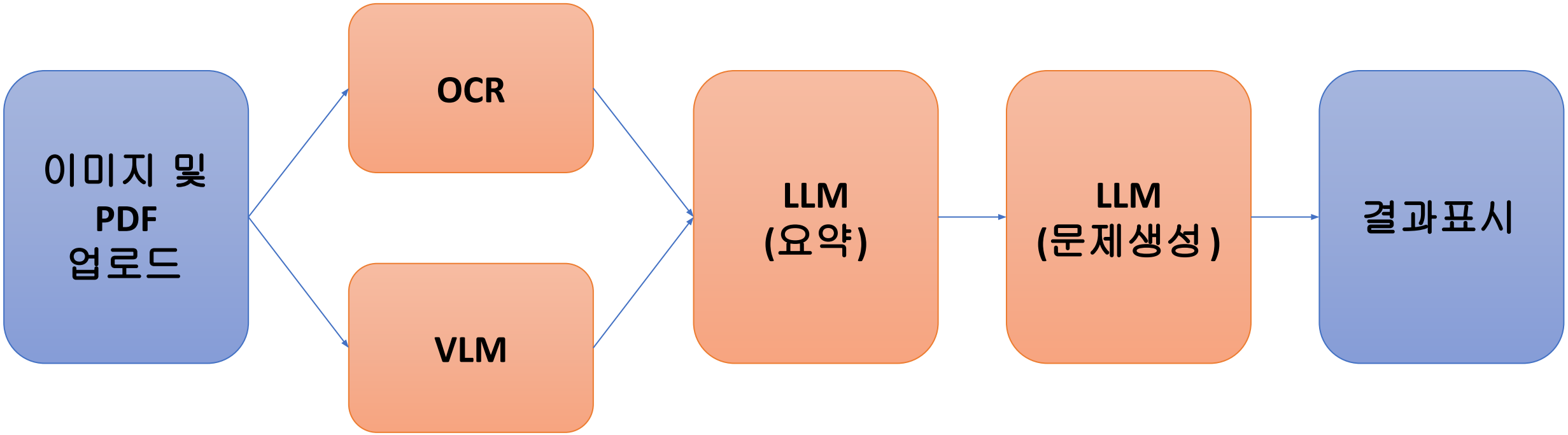
3. AI가 해당 내용 인식.

4. 요약.

5. 관련 문제 생성.

6. 결과 웹 UI 출력.

AI PROFESSOR - 흐름



구현 계획

- **OCR**: 이미지 속 자연어 인식.
- **VLM**: 이미지에서 텍스트 추출.
- **LLM**: 텍스트 요약, 퀴즈 형식 문제 생성.

Optical **C**haracter **R**ecognition

OCR - 구현 목표

이미지 or 문서 텍스트 추출.

- 이미지 및 PDF에서 자동 텍스트 추출.
- 한글 • 영문 혼합 추출.
- 텍스트가 없는 PDF >>> 자동 이미지 렌더링 후 OCR 수행.

OCR

```
def extract_text_from_pdf(path):
    ...
def extract_text_from_image(path):
    ...
```

- 각각 PDF, 이미지 파일을 처리하는 함수.

OCR - PDF

```
for page_num, page in enumerate(doc, start=1):

        text = page.get_text("text").strip()    # 먼저 텍스트 추출 시도.
        if text:
            full_text.append(text)              # 레이어가 있을 경우 바로 텍스트 추출.
        else:
                                                # 텍스트가 없는 경우 이미지로 변환하여 OCR 수행.
            try:
                pix = page.get_pixmap(dpi=300)  # DPI를 300으로 설정 >>> 해상도 • 인식률 향상.
                img = Image.open(io.BytesIO(pix.tobytes("png")))
                ocr = pytesseract.image_to_string(img, lang='eng+kor')
                full_text.append(f"[페이지 {page_num} OCR 결과]\n" + ocr.strip())
            except Exception as e:
                full_text.append(f"[페이지 {page_num} OCR 실패]: {str(e)}")

    return "\n\n".join(full_text)
```

OCR - 이미지

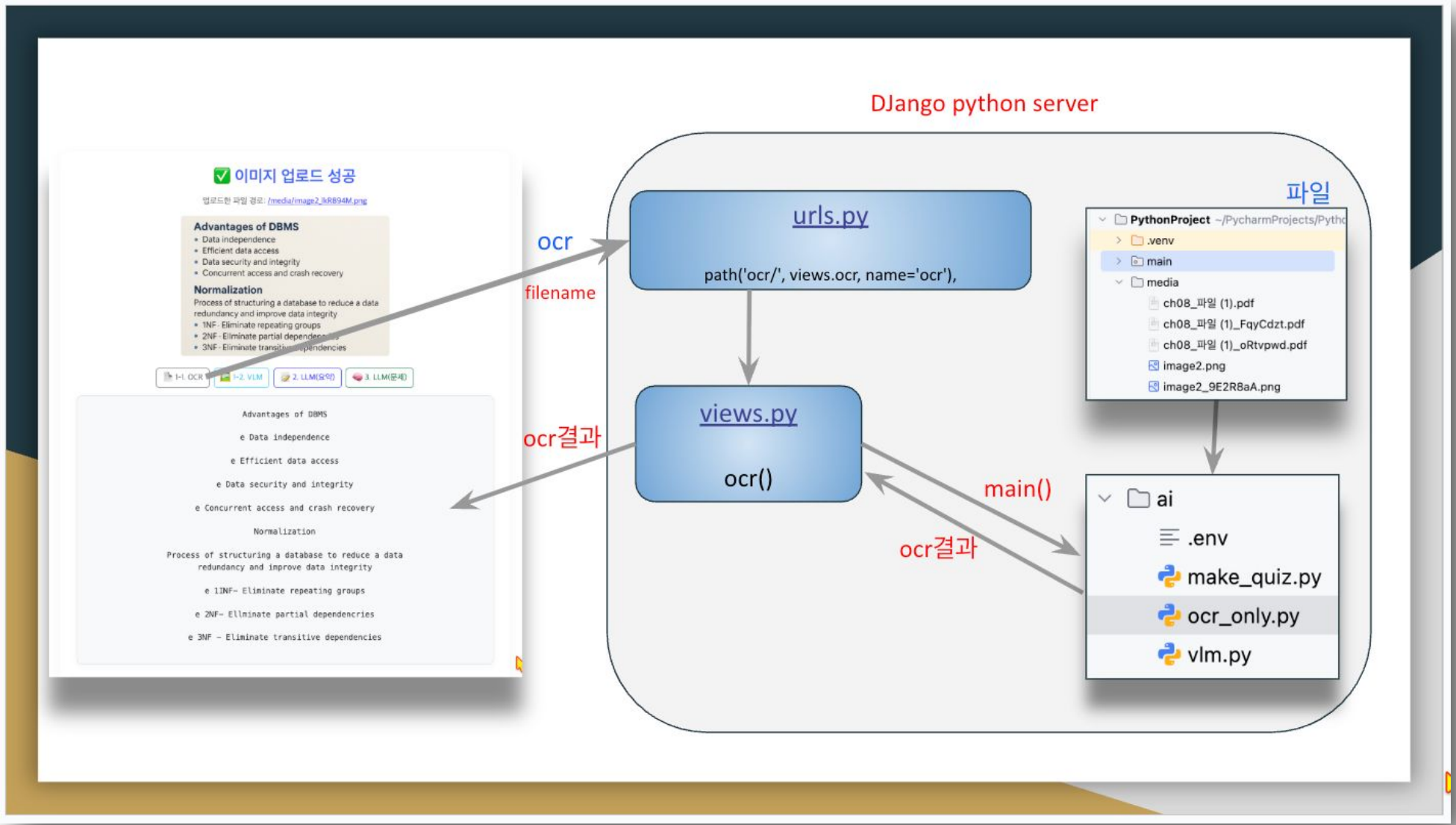
```
def extract_text_from_image(path):
    try:
        img = Image.open(path)
        text = pytesseract.image_to_string(img, lang='eng+kor')
        return text.strip()
    except Exception as e:
        return f"이미지 OCR 실패: {str(e)}"

#업로드된 이미지를 열고 OCR 적용.
```

OCR - 기술 스택

사용 모듈

- PyMuPDF: ‘PDF >>> 이미지’ 렌더링
- Pillow: 이미지 불러오기.
- Pytesseract: 텍스트 추출 - (광학 문자 인식) 파이썬 래퍼
- Tesseract OCR: 백엔드 OCR 엔진.
- eng+kor: 언어팩

Optical Character Recognition



OCR

- PDF / 이미지 자동 처리 >>> 멀티 포맷 지원.
- 텍스트 없을 때만 OCR을 적용 >>> 효율적.
- DPI 300 설정 >>> 해상도 튜닝.
- eng+kor 언어팩 >>> 한영 조합 가능.

Vision **L**anguage **M**odel

VLM - 구현 목표

이미지와 텍스트의 의미 이해

↓

텍스트 추출 (OCR 보조처리)

- 주 입력값이 “문서 및 이미지”.

>>> 초기 단계에서는 텍스트 추출이 중요.

VLM - 기술 스택

사용 모듈

- OpenCV-python: 이미지 전처리 .
- Pytesseract: 텍스트 추출(OCR 기술).
- Matplotlib: 시각화.

VLM

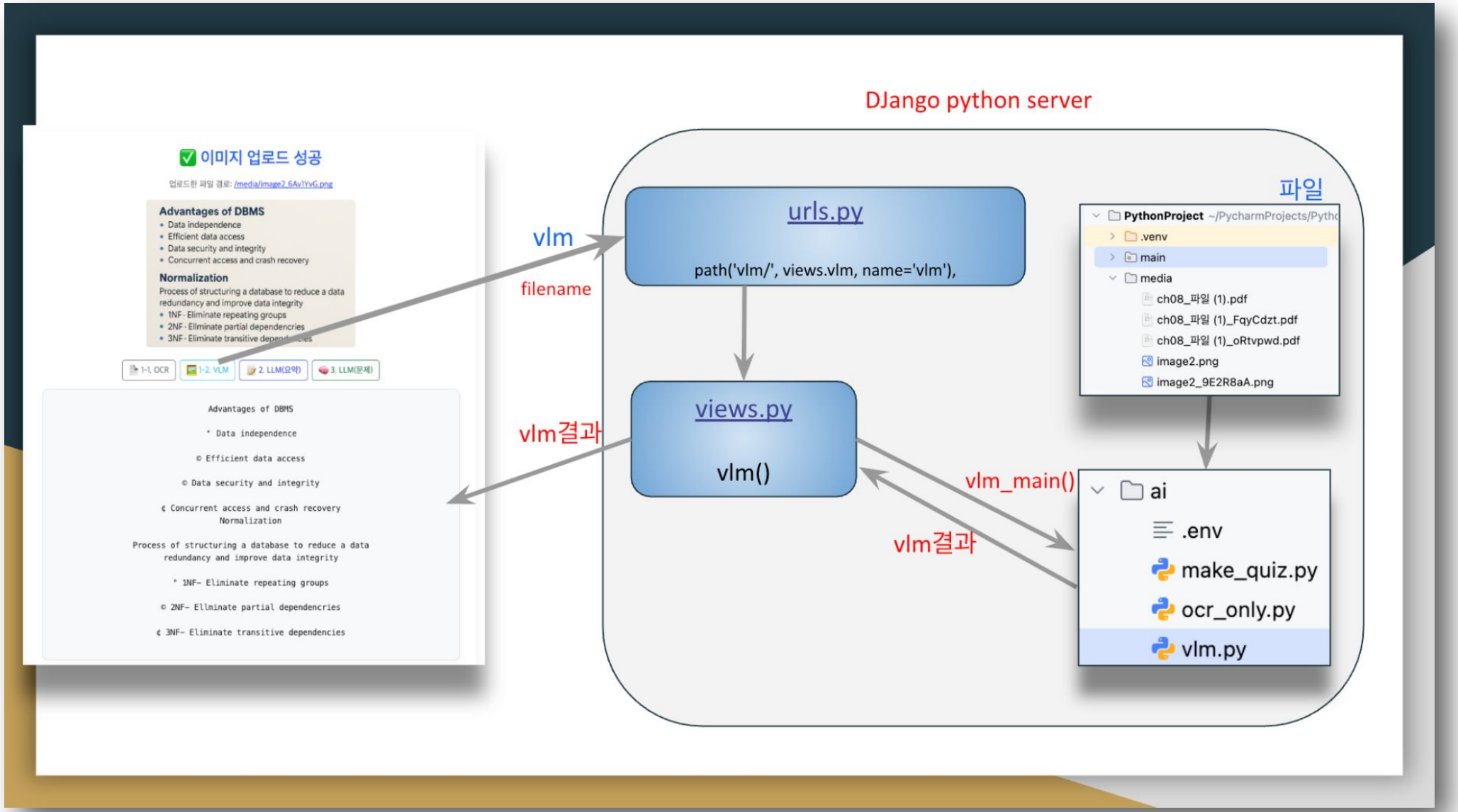
```
def vlm_main(image_file):
    image = cv2.imread(image_file)
    gray → blur → adaptiveThreshold 적용
    text = pytesseract.image_to_string( ... )
    return text


# OpenCV로 이미지 전처리, pytesseract로 텍스트 인식.
```

VLM

전처리 이미지 출력 + OCR과 결과 비교

- 이미지 전처리: Grayscale >>> Blur >>> Adaptive Threshold 이진화.
- Pytesseract: 최종 텍스트 추출.

Vision Language Model



Large **L**anguage **M**odel

LLM – 구현 목표

텍스트 요약 및 문제 생성

- 핵심 내용 요약
- OX / 객관식 문제 생성.

LLM

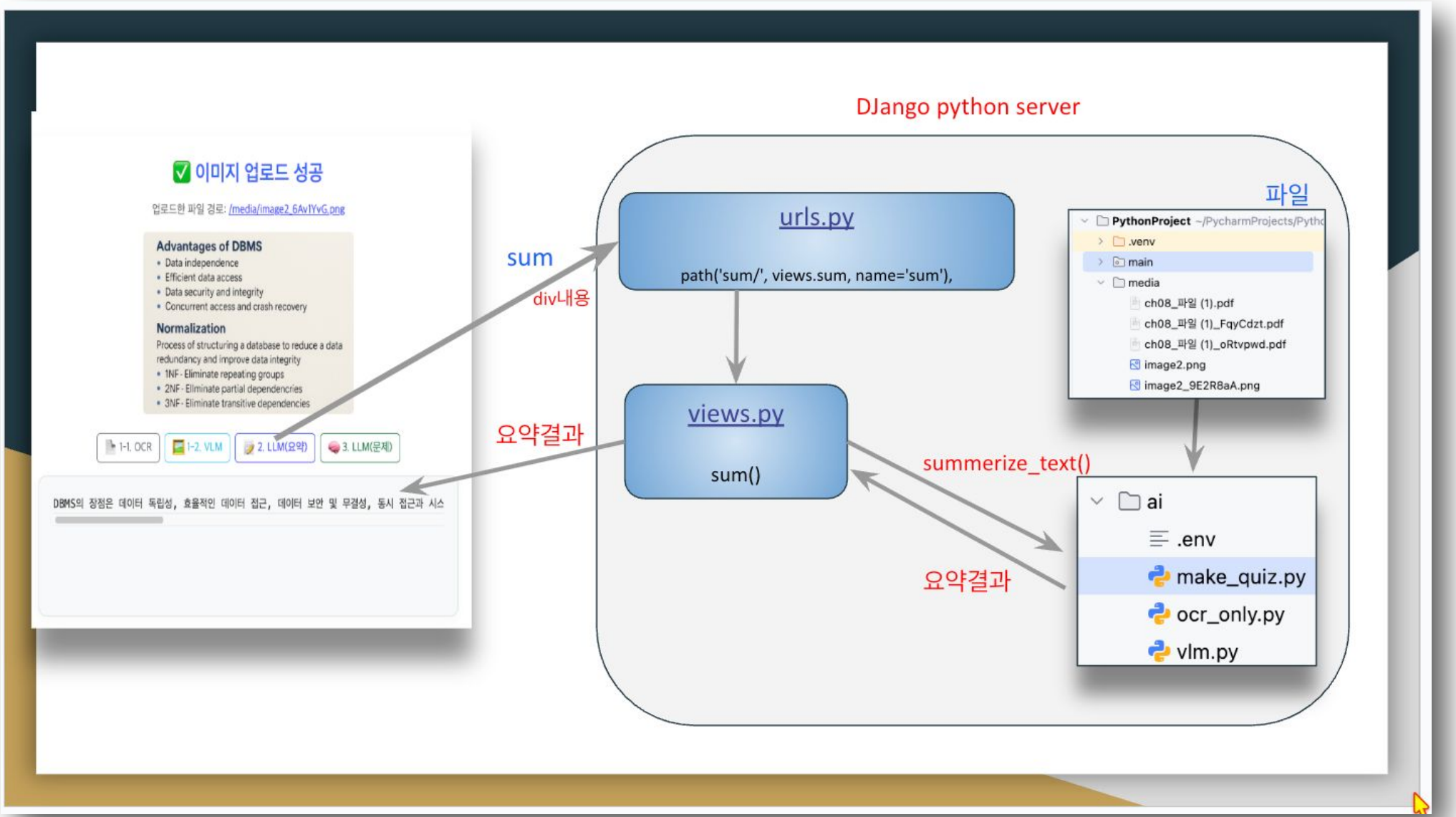
```
def summarize_text(text: str) → str:
    system_prompt = "교육 내용을 요약하는 AI입니다 ..."
    # ChatGPT API 호출 → 핵심 요약 반환

def generate_questions(text: str) → str:
    system_prompt = "문제 출제용 AI입니다 ..."
    # ChatGPT API 호출 → 퀴즈 3~6개 자동 생성
```

LLM - 기술 스택

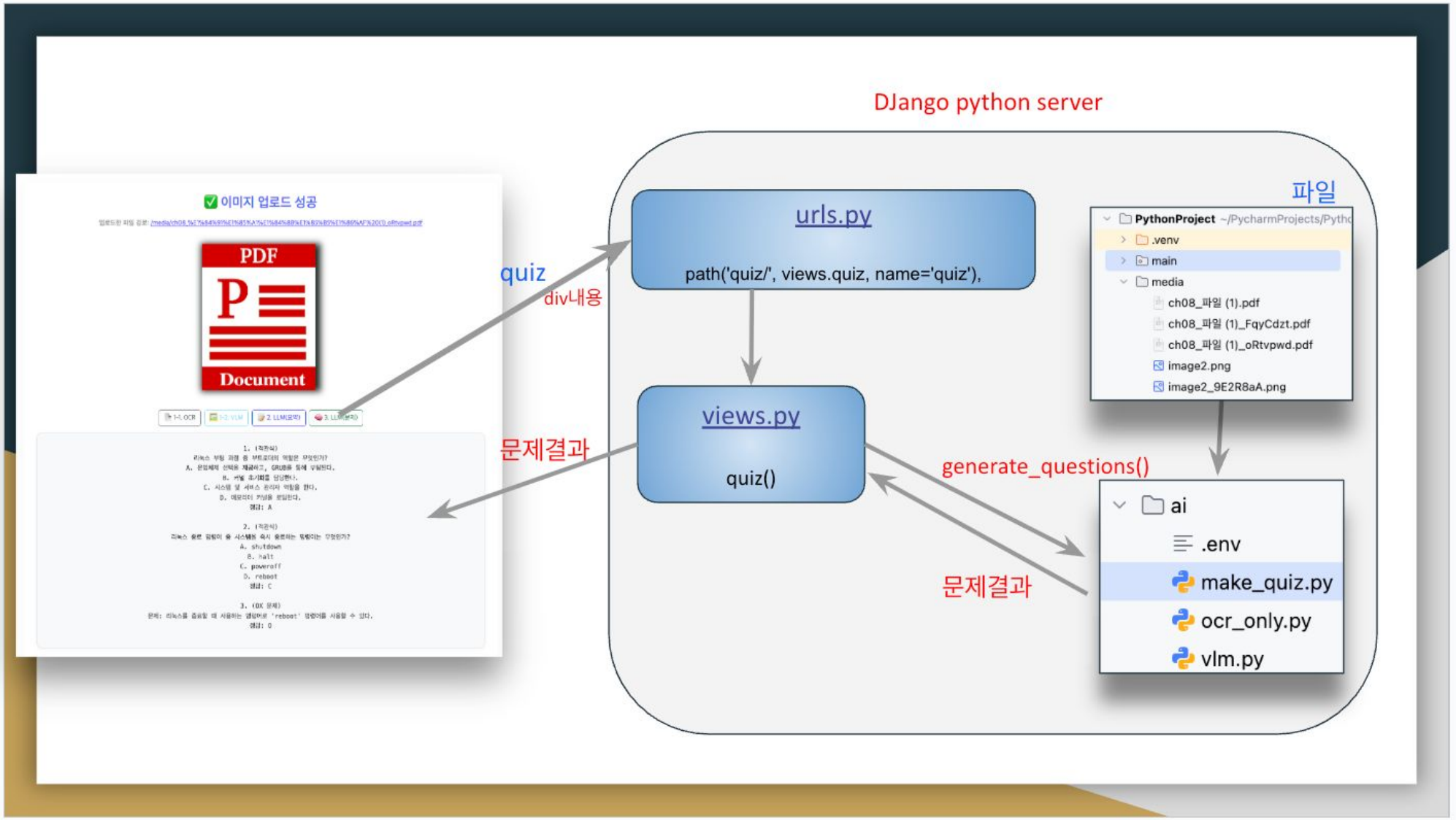
사용 모듈

- OpenAI GPT API
- .env 보안처리
- Python-dotenv

LLM - 요약


✅ 이미지 업로드 성공
업로드한 파일 경로: /media/image2_6Av1YvG.png
Advantages of DBMS
• Data independence
• Efficient data access
• Data security and integrity
• Concurrent access and crash recovery
Normalization
Process of structuring a database to reduce a data redundancy and improve data integrity
• 1NF - Eliminate repeating groups
• 2NF - Eliminate partial dependencies
• 3NF - Eliminate transitive dependencies
1-1. OCR
1-2. VLM
2. LLM(요약)
3. LLM(문제)
DBMS의 장점은 데이터 독립성, 효율적인 데이터 접근, 데이터 보안 및 무결성, 동시 접근과 시스
DJango python server
sum
div내용
urls.py
path('sum/', views.sum, name='sum'),
파일
PythonProject ~/PycharmProjects/Pytho
.venv
main
media
ch08_파일 (1).pdf
ch08_파일 (1)_FqyCdzt.pdf
ch08_파일 (1)_oRtvpwd.pdf
image2.png
image2_9E2R8aA.png
요약결과
views.py
sum()
summerize_text()
요약결과
ai
.env
make_quiz.py
ocr_only.py
vlm.py

LLM - 문제 제작


DJango python server
✅ 이미지 업로드 성공
PDF
Document
quiz
div내용
문제결과
urls.py
path('quiz/', views.quiz, name='quiz'),
views.py
quiz()
generate_questions()
문제결과
파일
PythonProject ~/PycharmProjects/Pytho
.venv
main
media
ch08_파일 (1).pdf
ch08_파일 (1)_FqyCdzt.pdf
ch08_파일 (1)_oRtvpwd.pdf
image2.png
image2_9E2R8aA.png
ai
.env
make_quiz.py
ocr_only.py
vlm.py

User **I**nterface / **U**ser e**X**perience

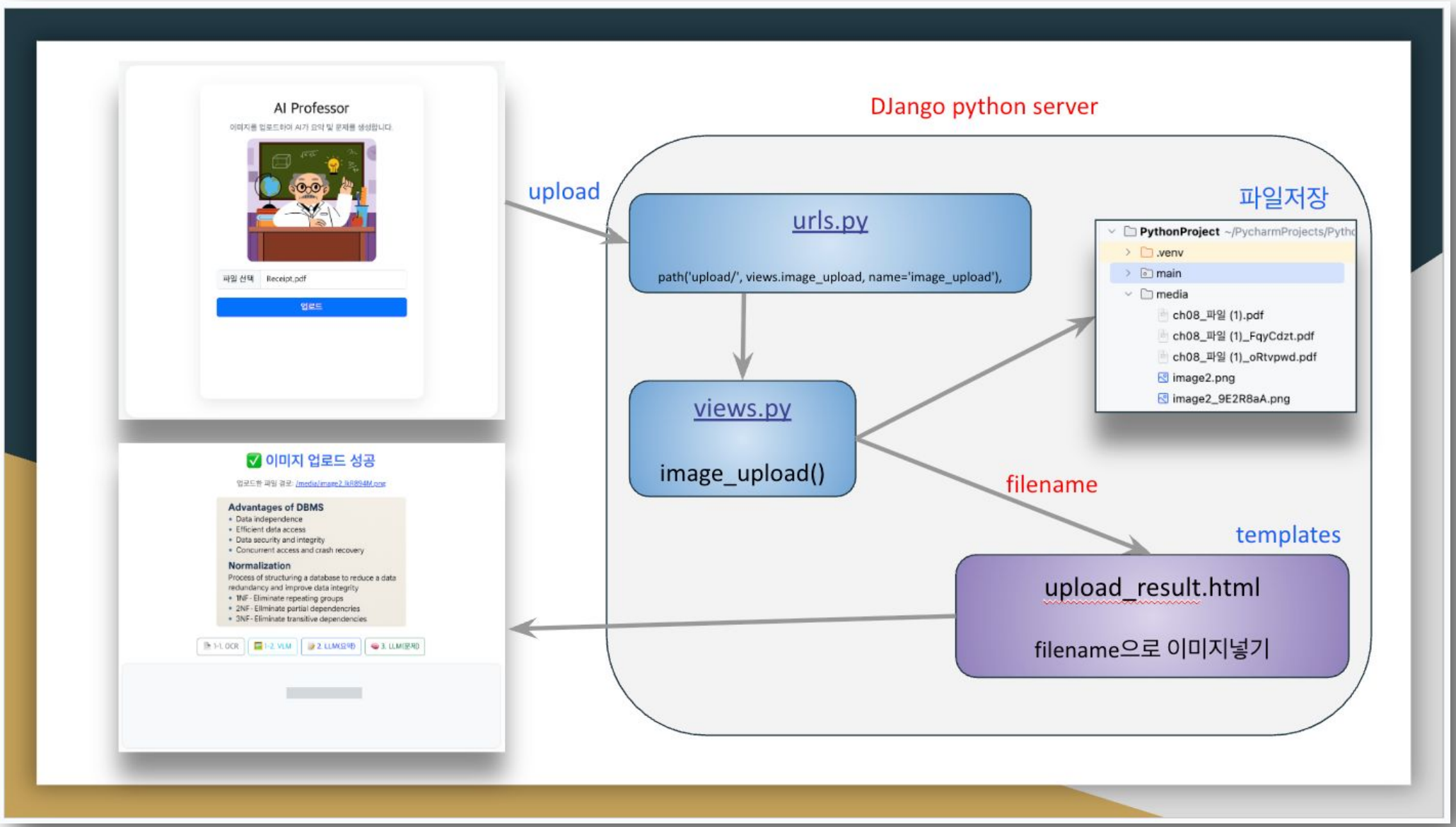
UI / UX- 구현 목표

- 이미지 업로드
- OCR/요약/문제
- 탭 전환, 입력창

UI / UX

- 웹 UI 구현: HTML/CSS/JavaScript 기반 인터페이스.
- 디자인 특징: 반응형 디자인 + 화면 리사이저 기능으로 비율 조절.
- 이미지 업로드 미확인시 알림 기능.

User Interface / User eXperience



확장 가능성

- 자동화 학습 미구현
- 텍스트 외 시각 자료
- 난이도 및 피드백 수준 증가