

차량용 AI 비서 시스템

20221003 공해성
20221035 이현경
20221042 허소영

목차

- 프로젝트(연구, 설계) 목표
- 계획대비 진행상황
- 역할 분담
- 연구 내용
- 중간 결과물
- 프로젝트 향후계획
- 참고 자료

1. 프로젝트(연구, 설계) 목표

- 차량 정보 데이터베이스를 구축해 음성/텍스트 입력을 통한 AI 응답 시스템을 개발
- 쇼핑몰 추천 시스템과 CO₂ 센서를 활용해 이산화탄소 농도를 감지 후 경고하는 기능 구현

차량 내 음성기반 AI 응답 + CO₂ 상태 감지 + 쇼핑 추천 기능을
통합한 스마트 비서 시스템 구축

1. 프로젝트(연구, 설계) 목표

지난 진행 내용 요약

- 데이터베이스 설계 및 구축
 - 차량정보, 경고등, 비상상황, 비상상황응급조치, 유지보수
- OpenAI API를 사용해 데이터베이스와 연결
- 데이터베이스 기반 답변기능 구현
 - 데이터베이스에 없는 정보는 답변 못함

1. 프로젝트(연구, 설계) 목표

1학기

1. 데이터베이스 설계
2. AI 응답 시스템
3. 음성/텍스트 처리
4. UI/UX 설계 및 구현

2학기

1. 쇼핑몰 추천 기능
2. CO2 감지 기능
3. 통합 테스트
4. 최종 보고서 작성

2. 계획대비 진행상황

☑ : 계획 발표 당시 설정한 예상 진행
 ➡ : 중간 발표 기준 실제 진행 상황
 ➡ : 기말 발표 기준 실제 진행 상황

연구 내용	3월	4월	5월	6월	7월	8월	9월	10월
주제 선정 및 계획서 작성	☑ ➡							
데이터베이스 설계 및 구축	☑ ➡	☑						
음성/텍스트 처리 구현		☑	☑ ➡					
AI 질문 응답 구현		➡	☑ ➡	☑				
UI/UX 설계 및 구현			➡	☑	☑			
쇼핑몰 연동					☑			
CO2 감지 기능 구현						☑		
통합 테스트 및 최적화							☑	☑
최종 보고서 작성								☑

3. 역할 분담

공해성	중간 발표 음성/텍스트 처리 구현
이현경	중간발표 자료작성 UI/UX 설계
허소영	AI 질문 응답 개발 README 작성

4. 연구 내용

□ AI 질문 응답 개발

- RAG 기반 GPT 응답 구조 구현
- 유사 질문 탐지 기능
- 질문 정규화
- 출처 명시 및 책임 회피 문구 자동 추가
- CLI 인터페이스 개선
- 코드 구조 리팩토링 및 파일 분리

4. 연구 내용

□ 음성/텍스트 처리 구현

- .env 파일에서 API 키 로딩
- 마이크로 음성녹음(wave) 저장
- Google Speech-to-Text로 음성인식 후 텍스트 변환, 출력
- DB에 내용 검색 후, 없을 시 GPT 응답 생성
- 설명/해결/응급조치 단계만 추출해 TTS로 음성 출력
- 질문과 응답 chat_log 테이블에 저장

4. 연구 내용

□ UI/UX 설계

- 홈화면 : 현재 시간, CO₂ 수치 및 상태 아이콘 표시
음성 인식 버튼 → 질문 입력 유도
- AI 대화 화면 : 사용자의 질문과 시스템 응답을 표시
대답은 출처/책임 문장과 함께 제공
- CO₂ 화면 : 15분 간격/일간/월간 CO₂ 그래프 슬라이드 구성
색상별 상태 아이콘(좋음, 보통, 나쁨, 위험)


▶ 쇼핑몰 화면은 추후 구현 결정 후 제작

5. 중간 결과물

AI 질문 응답

```
def normalize_question(text):  
    text = text.lower()  
    text = re.sub(r"^[^\w\s]", "", text) # 특수문자 제거  
    text = re.sub(r"\s+", " ", text).strip()  
    return text
```

유사도 검사

“엔진오일 경고등”
“엔진오일 경고등이 뭐야?”  같은 질문으로 처리

5. 중간 결과물

AI 질문 응답

```
def find_similar_answer(normalized_question, car_model):
    conn = get_connection()
    cursor = conn.cursor()

    # 모든 질문 불러와서 유사도 확인
    cursor.execute("SELECT question, answer FROM solution WHERE car_model = %s", (car_model,))
    for q, a in cursor.fetchall():
        if is_similar(q, normalized_question):
            conn.close()
            return a, "solution", q

    conn.close()
    return None, None, None
```

- GPT 비용 절감을 위해 이미 유사한
질문이 있는 경우 재활용

5. 중간 결과물

AI 질문 응답

```
def save_answer(question, answer, car_model):  
    conn = get_connection()  
    cursor = conn.cursor()  
    cursor.execute(  
        "INSERT INTO solution (question, answer, car_model) VALUES (%s, %s, %s)",  
        (question, answer, car_model)  
    )  
    conn.commit()  
    conn.close()
```

- GPT가 생성한 답변은 DB에 저장하여
다음 질문에 활용 가능

5. 중간 결과물

AI 질문 응답

```
def search_all_tables(keyword):
```

```
    conn = get_connection()
```

```
    cursor = conn.cursor()
```

```
    queries = [
```

```
        ("경고등", """
```

```
            SELECT '경고등' AS category, warning_name, warning_desc, solution
```

```
            FROM warning_lights
```

```
            WHERE warning_name ILIKE %s OR warning_desc ILIKE %s OR solution ILIKE %s
```

```
            LIMIT 1
```

```
        """, (f"%{keyword}%", f"%{keyword}%", f"%{keyword}%")),
```

- RAG 기반 응답 방식 구현

- search_all_tables()는 키워드로 DB를 검색하여 GPT에게 참고 문맥으로 제공

- 테이블에 대한 각각의 쿼리를 작성

5. 중간 결과물


AI 질문 응답


```
response = openai.chat.completions.create(  
    model="gpt-3.5-turbo",  
    messages=[  
        {"role": "system", "content": f"당신은 차량 AI 정비 도우미입니다. {context}"},  
        {"role": "user", "content": question}  
    ]  
)
```


문장을 생성할 때 **차량에 대한 답변만** 하도록 역할을 부여해
정확하고 매끄러운 문장을 작성하도록 함

5. 중간 결과물


AI 질문 응답 - 실행 결과(DB)

 차량 모델을 입력하세요 : 아반떼

 질문 : 브레이크 경고등

 DB 답변 (출처 : solution) [Q: 브레이크 경고등]

브레이크 경고등이 켜져 있다면, 브레이크액의 양을 먼저 확인해보시는 것이 좋습니다. 브레이크액이 부족하다면 보충해야 합니다. 그러나 보충 후에도 브레이크 경고등이 계속 켜져 있을 경우에는 당사 직영 하이테크센터나 블루핸즈에서 전문가에게 점검을 받으시는 것이 좋습니다. 위험을 미리 예방하기 위해 브레이크 시스템을 정기적으로 점검하는 것이 중요합니다. 추가 도움이 필요하시면 언제든지 물어보세요.

 참고 : 이 정보는 일반적인 설명이며, 실제 차량 매뉴얼도 확인하세요.

 GPT 보완 설명도 들을까요? (Y/N): N

실행 흐름 : 차량 모델 입력 - 질문 입력 - 답변 (DB) - 보완 설명 여부

5. 중간 결과물

AI 질문 응답 - 실행 결과(GPT)

? 질문 : 노란색 경고등

💬 GPT 생성 응답 :

노란색 경고등이 켜진 경우에는 주행 중에 문제가 발생했음을 나타내므로 즉시 안전한 장소에 세워서 엔진을 멈추고 엔진을 확인해야 합니다. 경고등이 어떤 이유로 켜졌는지 정확히 파악하기 위해서는 전문가의 도움이 필요할 수 있습니다. 가능하다면 가까운 정비소를 방문하여 문제를 진단할 수 있는 진단을 받는 것이 좋습니다.

🔍 참고 : 이 정보는 일반적인 설명이며, 실제 차량 매뉴얼도 확인하세요.

? 질문 : 종료

👋 종료합니다. 안전 운전 하세요!

실행 흐름 : 차량 모델 입력 - 질문 입력 - 답변 (GPT) - 종료 (질문창에 '종료' 입력)

5. 중간 결과물

음성/텍스트 처리

```
print("🎤 말해주세요...")
p = pyaudio.PyAudio()
stream = p.open(format=FORMAT, channels=CHANNELS, rate=RATE,
                 input=True, frames_per_buffer=CHUNK)

frames = []
for _ in range(0, int(RATE / CHUNK * seconds)):
    data = stream.read(CHUNK)
    frames.append(data)

stream.stop_stream()
stream.close()
p.terminate()
```

사용자의 음성을 **실시간**으로 읽어와,
이후 처리에 사용할 오디오 데이터로 **저장**

5. 중간 결과물

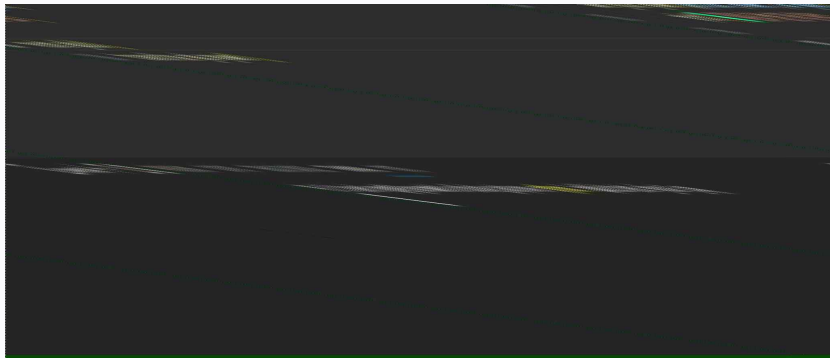
음성/텍스트 처리

```
if not answer:
    print("🗣️ GPT 호출 중...")
    gpt_response = openai.chat.completions.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "system", "content": f"You are a car assistant for model: {car_model}."},
            {"role": "user", "content": question}
        ]
    )
    answer = gpt_response.choices[0].message.content.strip()
    print("💬 GPT 응답:")
```

- DB에서 관련 응답이 없을 경우, OpenAI GPT를 호출해 질문에 대한 답변을 생성
- 차량 모델 정보를 시스템 프롬프트로 함께 전달하여 차량에 맞는 응답을 유도

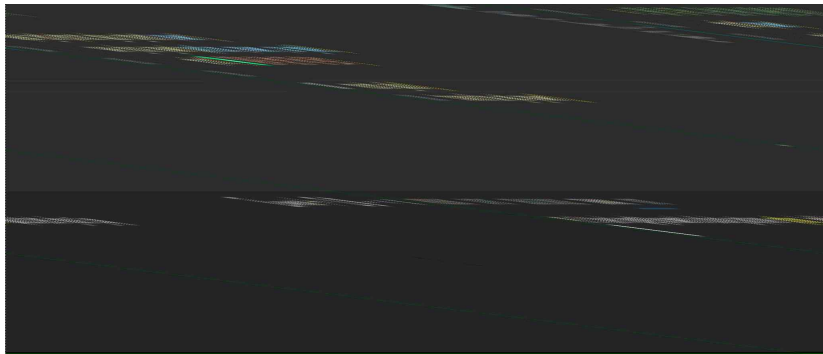
5. 중간 결과물

음성/텍스트 처리 - 실행영상(DB)



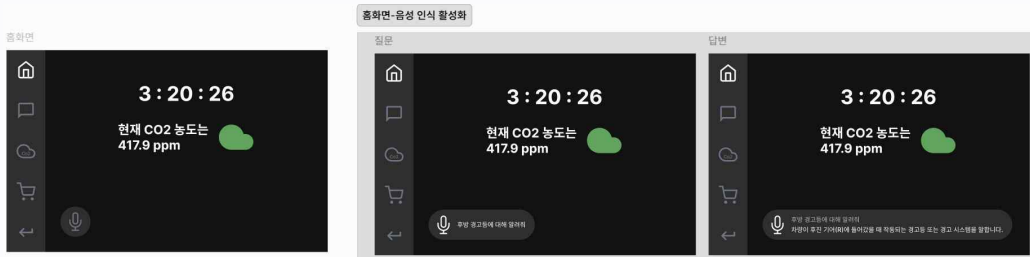
5. 중간 결과물

음성/텍스트 처리 - 실행영상(GPT)



5. 중간 결과물

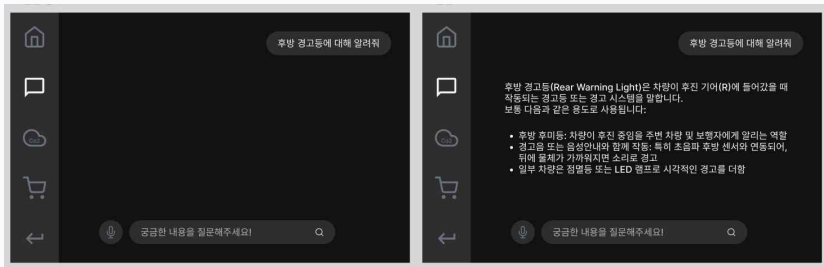
UI/UX 설계



- 현재 CO₂ 수치와 상태를 아이콘으로 표시
- 음성 인식 버튼을 통해 질문 기능으로 연결

5. 중간 결과물

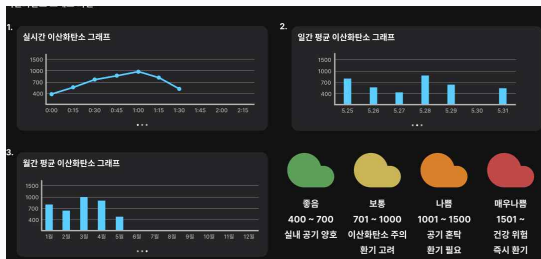
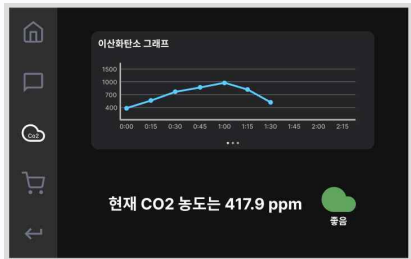
UI/UX 설계



사용자의 음성 또는 텍스트 입력에 따른 응답 출력

5. 중간 결과물

UI/UX 설계



슬라이드 넘김으로 15분 간격/일간/월간 CO₂ 그래프 제공

6. 프로젝트 향후계획

☑ : 계획 발표 당시 설정한 예상 진행

➡ : 향후 진행 예정인 작업 계획

연구 내용	3월	4월	5월	6월	7월	8월	9월	10월
주제 선정 및 계획서 작성	☑							
데이터베이스 설계 및 구축	☑	☑						
음성/텍스트 처리 구현		☑	☑	➡				
AI 질문 응답 구현			☑	☑	➡			
UI/UX 설계 및 구현				☑	➡			
쇼핑몰 연동					➡	➡		
CO2 감지 기능 구현					➡	➡		
통합 테스트 및 최적화							➡	☑
최종 보고서 작성								➡

7. 참고 자료

- ❑ 네이버blog - 음성인식 실습 파이썬 (Speech Recognition, PyAudio)
- ❑ PyAudio: Cross-platform audio I/O for Python, with PortAudio
- ❑ brunch story - 다크모드의 UI 컬러
- ❑ Tistory - UI/UX : 다크모드 UI 디자인 원칙 및 적용 가이드
- ❑ github - 라즈베리파이에서의 pyAudio 모듈 사용 예제

Q&A

감사합니다