SK네트웍스 Family AI과정 3기

모델링 및 평가 시스템 아키텍처

□ 개요

- 산출물 단계 : 모델링 및 평가
- 평가 산출물 : 시스템 아키텍처
- 제출 일자 : 12/30
- 깃허브 경로 : https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN03-FINAL-6Team
- 작성 팀원 : 최연규

□ 시스템 설계 목표 및 전략

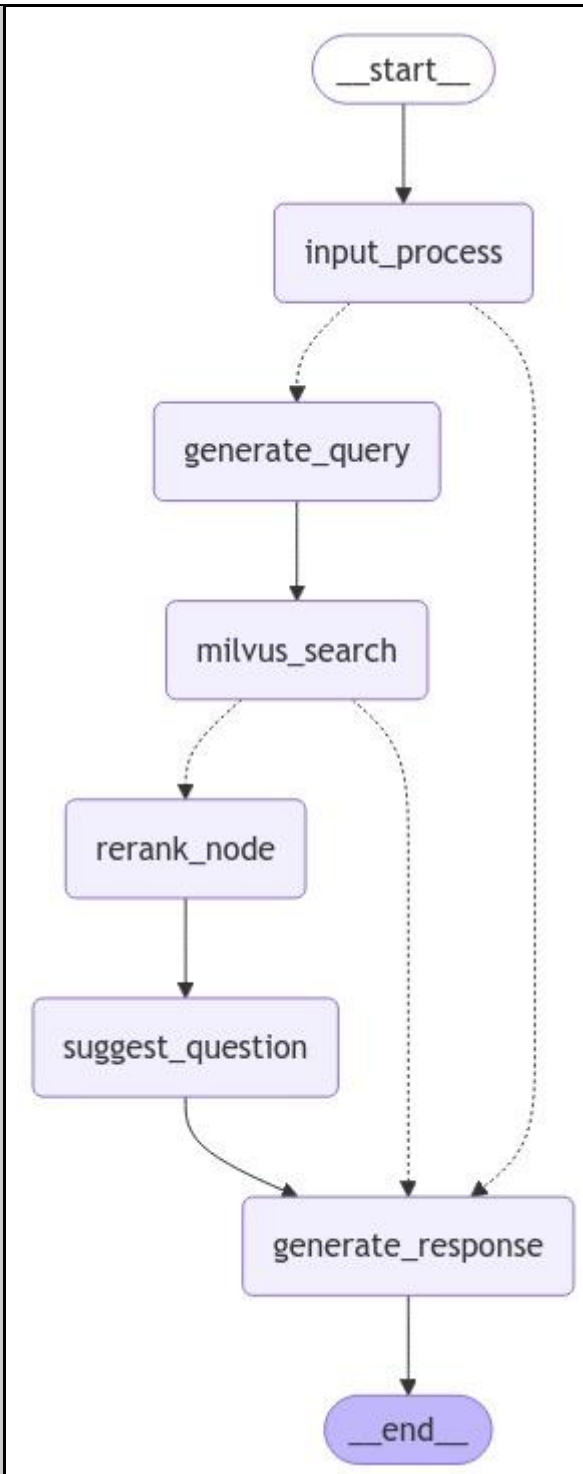
차량 추천, 차량 매뉴얼 Q&A, 보험 문의의 세 가지 주요 기능을 구현하기 위해 설계

□ 사용 기술 스택

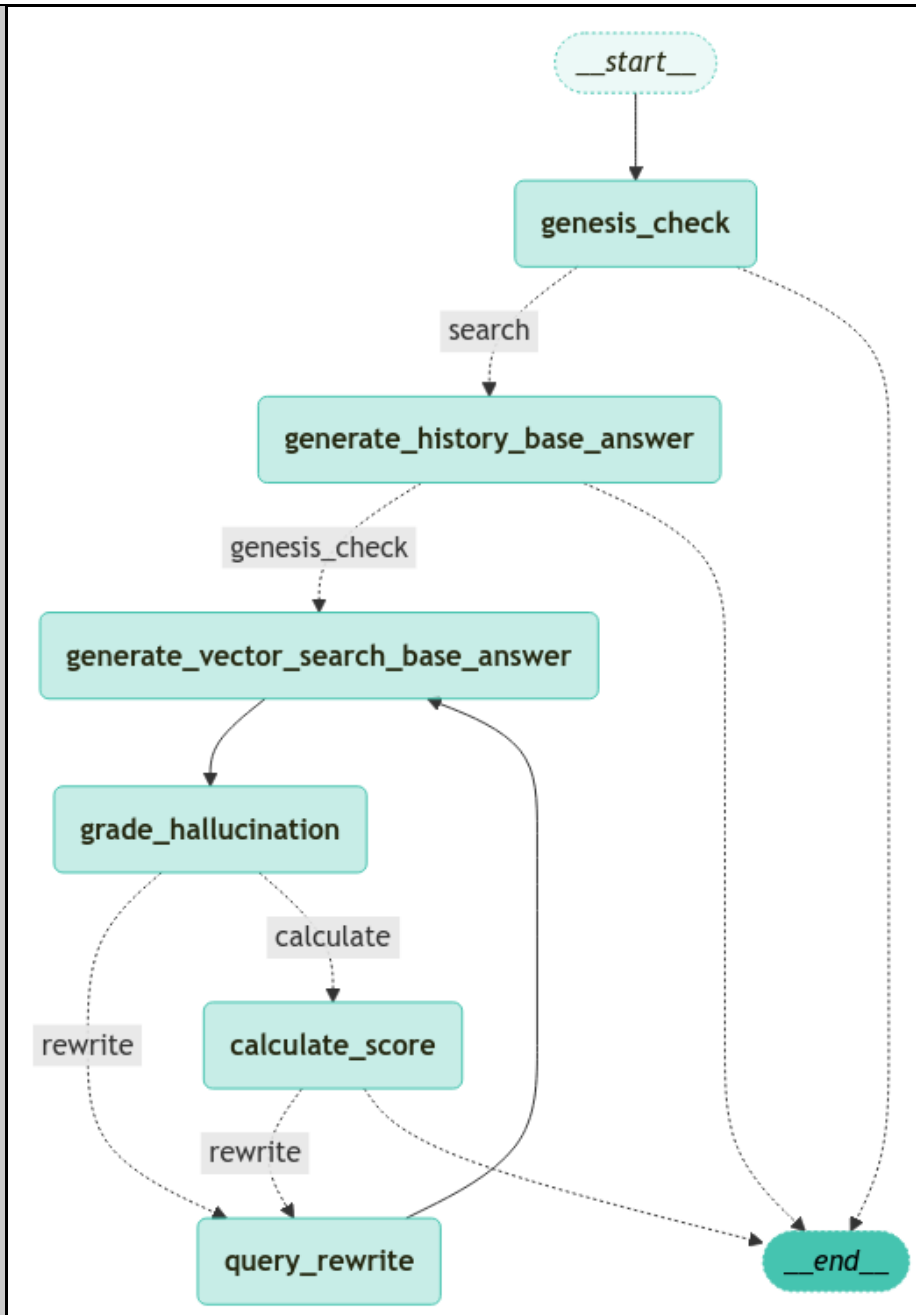
- 백엔드: Python (FastAPI)
- 벡터 검색: Milvus
- 데이터베이스: Mysql
- 모델링: OpenAI GPT, KoGPT
- 리랭커 모델: bge-reranker-v2-m3
- 임베딩 모델: bge-m3
- 배포 및 관리: AWS (ECR, RDS)

□ 기능 별 구성 요소

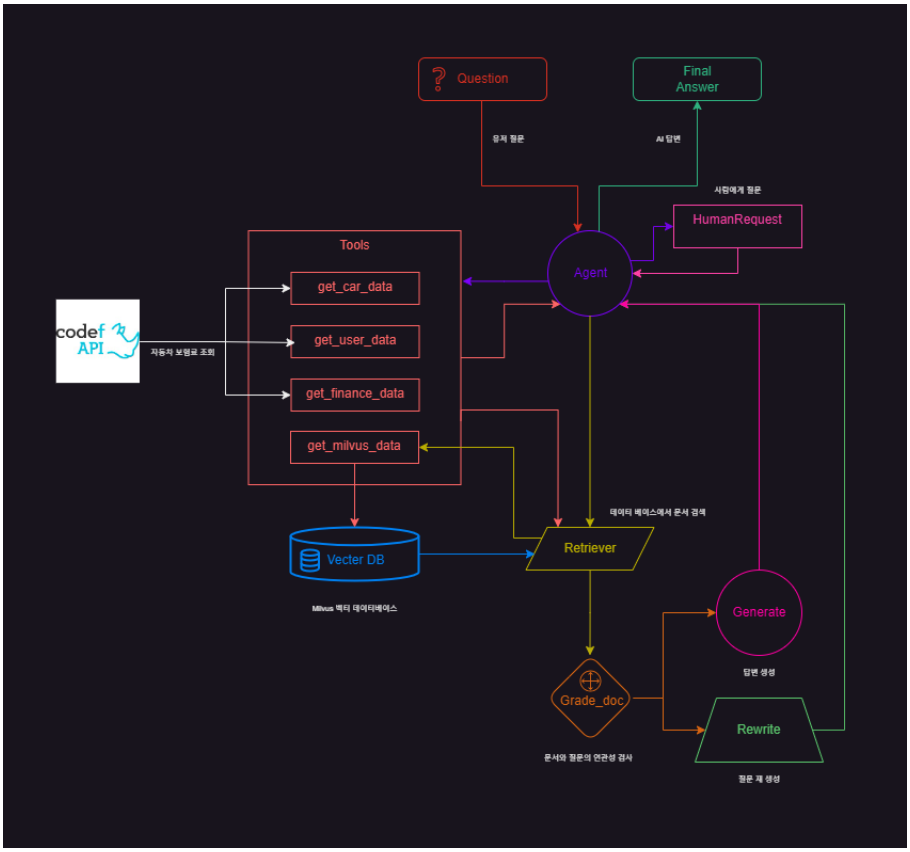
차량추천	
개요	사용자 질문에 기반한 차량 추천 서비스를 제공하기 위해 설계하였습니다. 핵심 목표는 사용자 질문을 이해하고, 데이터베이스에서 적합한 차량 정보를 검색한 후, 최적화된 추천을 제안하는 것입니다.
구성 요소	● **Input Process**: 사용자 입력을 처리하여 적절한 데이터 형식으로 변환합니다. ● **Generate Query**: 전처리된 입력 데이터를 기반으로 데이터베이스에서 검색할 수 있는 쿼리를 생성해 입력한 조건에 맞는 차량 목록을 읽어옵니다. ● **Milvus Search**: 사용자 후기가 저장된 Milvus 벡터 데이터베이스에서 사용자의 입력과 가장 유사한 결과를 검색해 차량 목록을 읽어옵니다 ● **Rerank Node**: 검색되어진 결과를 재정렬하여 사용자 의도에 가장 적합한 결과를 상위에 배치합니다. ● **Suggest Question**: 사용자 질문이 불명확하거나 추가 정보가 필요한 경우, 적합한 후속 질문을 제안합니다. ● **Generate Response**: 재정렬된 데이터를 기반으로 사용자 질문에 대한 최종 응답을 생성합니다.

데이터 흐름	
차량 매뉴얼 Q&A	
개요	차량 매뉴얼 데이터를 기반으로 사용자의 질문에 대한 정확하고 유용한 답변을 제공하기 위해 설계했습니다.

구성 요소	● **Genesis Check**: 초기 사용자 질문을 분석하여 제네시스 차량 관련 질문인지 판단합니다.. ● **Generate History-Based Answer**: 과거 대화 기록을 참조하여 대답이 가능한지 판단하여 답변을 생성합니다. ● **Generate Vector Search-Based Answer**: 사용자의 질문을 벡터화하여 Milvus 벡터 데이터 베이스에서 하이브리드 서치 후 리랭크하여 추출된 context를 기반으로 답변을 생성합니다. ● **Grade Hallucination**: 생성되어진 답변에 할루시네이션이 있는지 확인합니다. 생성된 답변이 이상이 없다면 점수 측정 노드로 이동하고 이상이 있다면 쿼리 재생성 노드로 이동합니다. ● **Calculate Score**: 생성된 답변과 사용자의 질문의 관련성과 정확성을 점수화 하여 평가합니다. 점수가 낮다면 쿼리 재성성 노드로 이동합니다. ● **Query Rewrite**: 질문을 답변이 잘 나오도록 재작성하여 개선된 검색 결과를 유도합니다. 재작성된 결과는 다시 **Generate Vector Search-Based Answer**로 전달합니다. ● **종료 (End)**: 최종적으로 생성된 답변을 사용자에게 제공하며 프로세스 종료합니다.

데이터 흐름	 __start__ genesis_check search generate_history_base_answer genesis_check generate_vector_search_base_answer grade_hallucination calculate rewrite calculate_score rewrite query_rewrite __end__
보험 문의	

개요	사용자가 가입한 보험에 대한 복잡한 정보를 질문 할 수 있는 챗봇 생성을 목적으로 합니다. 또 한 선택한 차량과 사용자의 정보를 토대로 차량 구매 시 산정 될 보험료에 대한 정보를 조회할 수 있습니다.
구성 요소	● **Input Process**: 사용자 입력을 처리하여 적절한 데이터 형식으로 변환합니다. ● **Agent**: 전처리된 입력 데이터를 기반으로 사용자의 질문 의도를 파악하고 다음 행동을 결정합니다. 다음 행동으로는 도구를 이용한 API호출, 사람에게 재 질문, 문서 검색 등이 있습니다. ● **Tools**: API를 호출하여 정보를 받아오거나, 데이터베이스의 문서를 검색합니다. ● **Retriever**: 사용자 질문을 토대로 의도에 맞는 약 3-5의 질문으로 재 생성 후 Milvus 벡터 데이터베이스에서 사용자의 입력과 가장 유사한 결과를 검색해 보험 약관 문서를 읽어옵니다. ● **Grade_docs**: 검색되어진 문서와 사용자의 첫 질문과의 연관성을 확인하여 답변 생성으로 이동할지, 질문 재 생성으로 이동할지 결정합니다. ● **Rewrite**: 사용자 질문과 참고 문서의 연관성이 적절하지 못하다면 질문을 첫 번째 질문을 토대로 질문을 재 생성 해 Agent로 이동합니다. ● **Generate Response**: 사용자 질문과 참고 문서의 연관성이 적절하다면 해당 문서를 토대로 사용자 질문에 맞는 최종 답변을 생성합니다. ● **HumanRequest**: 사용자 질문이 모호하거나 사용자의 입력이 필요한 작업은 사용자에게 질문을 통해 정보를 업데이트 합니다.

데이터 흐름	

□ 모듈 간 상호작용 및 통합성

- 응답 데이터의 스키마를 표준화

```python
class ChatResponse(BaseModel):  2개의 사용 위치  mabasa
    response: str = Field(..., description="AI의 최종 응답 텍스트")
    session_id: str = Field(..., description="현재 세션 ID")
    agent_id: str = Field(..., description="기능 설명")
    response_type: str = Field(..., description="응답 유형 (e.g., 사용자, ai)")
    page_info: dict = Field(None, description="도구에서 반환된 key-value 결과")
    suggest_question: list = Field(..., description="예상 질문")
    timestamp: datetime = Field(..., description="응답 생성 시각")
```