프로젝트 명세서

# 1. 프로젝트 개요

• 팀명: 하루하루  
• 프로젝트 이름: Advanced Project - 챗봇 개발  
• 목표 및 개요: 사용자의 자연어 입력에 대해 4가지 주제(축제정보, 스팀게임정보, 주식, 게임 아이템 경매장)에 대한 정보를 제공하는 챗봇 제작  
• 주요 기능 요약: 사용자 질문을 분석해 인텐트를 분류하고, 각 주제에 맞는 데이터를 수집 및 가공하여 자연어 응답으로 제공

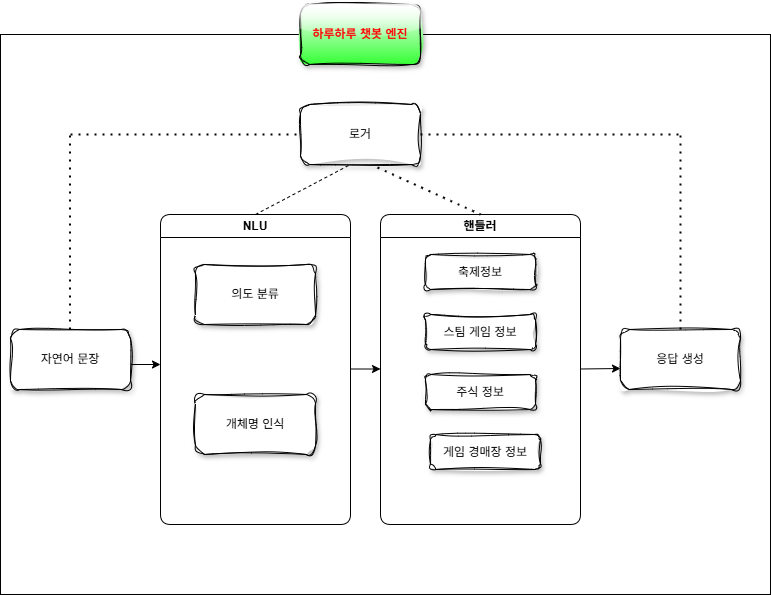
# 2. 기획 배경 및 목적

• 문제 정의: 다양한 주제의 정보를 일일이 검색하지 않고, 간단한 대화형 인터페이스로 확인할 수 있도록 지원  
• 사용자 시나리오: 사용자가 자연어로 "요즘 축제 뭐 있어?" 라고 입력 → 챗봇이 D에 저장된 축제 정보를 수집해 응답  
• 필요성과 기대 효과: 정보 접근성 향상, 주제별 통합 정보 제공, 사용자 친화적 인터페이스

# 3. 기술 스택

• 프론트엔드: JavaScript, React  
• 백엔드: Python (FastAPI, uvicorn)  
• NLU 엔진: Python 기반 intent 분류기 직접 구현  
• 데이터 수집: Python (공공데이터 API 활용)  
• QA 및 학습용 데이터: JSON, Pandas  
• 협업 도구: GitHub, Discord, KakaoTalk

# 4. 시스템 아키텍처

• 엔진: 사용자 질문 분석(intent, entity 분류) 및 각 핸들러별 응답생성   
• 프론트엔드: React로 사용자 입력 및 응답 표시 UI 구성

• 백엔드: FastAPI 기반 서버, 인텐트 처리 및 데이터 반환

• 외부 연동: 한국투자증권API, NC개발자API

# 5. 주요 기능 및 세부 설명

|  |  |  |  |
| --- | --- | --- | --- |
| 기능 이름 | 설명 | 관련 화면/엔드포인트 | 담당자 |
| 축제 정보 조회 | 공공데이터 API를 통해 현재 축제 정보 응답 | /festival | 최주호 |
| 스팀 게임 검색 | 게임명으로 스팀 게임 정보 제공 | /steam | 정원민 |
| 주식 정보 제공 | 종목명을 통한 시세 정보 반환 | /stock | 정원민 |
| 게임 아이템 경매 정보 | 경매장 시세 정보 응답 | /auction | 정원민 |

# 6. API 명세

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | Endpoint | 설명 | Request 예시 | Response 예시 |
| GET | /festival?region=서울 | 서울 지역 축제 검색 | - | { "festivals": [...] } |
| GET | /steam?title=elden ring | 게임명으로 스팀 게임 검색 | - | { "game\_info": {...} } |
| GET | /stock?ticker=005930 | 주식 종목 정보 검색 | - | { "price": 123456 } |
| GET | /auction?item=검은칼날 | 게임 아이템 경매 시세 | - | { "price": 1000000 } |

# 7. 데이터 포맷 예시

예시 JSON:  
{  
 "intent": "search\_festival",  
 "entities": {  
 "region": "서울"  
 }  
}

# 8. QA 및 테스트 방법

• 테스트 방식: 시나리오 기반 QA, API 호출 테스트  
• 테스트 도구: Postman, 직접 시나리오 QA 진행  
• 오류 예외 처리: 입력 누락, 인텐트 미매칭, API 실패 처리

# 9. 역할 분담

|  |  |
| --- | --- |
| 이름 | 역할 |
| 소형섭 | NLU 엔진 및 인텐트 핸들러 |
| 최주호 | 공공데이터 API 수집 |
| 전현준 | 백엔드 개발 |
| 정원민 | 학습용 데이터 제작 및 QA |
| 신제윤 | 프론트엔드 제작 및 QA |

# 10. 추가 사항

• 회의록 링크: [URL 또는 문서 위치]  
• 향후 개선 방향: 사용자 맞춤 추천 기능, 챗봇 학습 자동화, 다국어 지원, Login/out 기능 DB에 User정보 삽입후 정보 저장  
• 배포 링크 / 실행 방법: [서버 주소 또는 실행 방식 기술]