

## 1.0 공통 기능

- **Generative AI, RAG, Vision, ML, FAQ, ABOUT US**, 로그인/회원가입 으로 네비게이션 메뉴가 구성되어야 한다.
- 각 메뉴는 클릭을 통해 해당 페이지로 이동할 수 있어야 한다.
- 로고 클릭 시, 메인화면으로 이동할 수 있어야 한다.
- 반응형 구조를 적용하여 화면 크기가 일정 이하일 경우 네비게이션이 햄버거 메뉴 형태로 축소되어야 한다.
- 모든 페이지는 반응형 구조를 지원하여 **PC,태블릿,모바일** 환경에서 정상적으로 표시되어야 한다.
- 네비게이션 메뉴에 마우스를 올리면 해당 메뉴의 색상이 변화해야 한다.
- 헤더와 푸터는 모든 페이지에서 동일한 구조를 가져야 한다.
- 페이지 하단에는 **Github** 링크가 제공되어야 하며, 클릭 시 프로젝트 저장소 페이지로 이동할 수 있어야 한다.
- 모든 기능 페이지는 로딩 중일때 로딩 상태를 시각화해서 사용자가 진행 중임을 인지할 수 있도록 한다.
- 모든 모델 응답은 한국어를 기본으로 제공해야 한다.

## 1.1 메인 페이지 기능

- 메인 페이지 맨 위쪽에는 케러셀과 페이지 버튼들이 위치하며 클릭시 해당 페이지로 이동할 수 있어야한다.
- 케러셀에는 **Generative AI, RAG, Vision, ML** 페이지로 구성되어 있어야 한다.
- 메인페이지 중앙에는 서비스 소개글이 요약되어서 나오며 마우스를 올리면 움직일 수 있어야 한다.
- 메인페이지 하단에는 사용자 시나리오 박스와 페이지 버튼들이 위치하며 사용자가 시나리오를 보고 어떤 페이지인지 알 수 있어야하며 버튼을 클릭시 해당 페이지로 이동할 수 있어야 한다.
- 시나리오 박스는 **Generative AI, RAG, Vision, ML** 페이지로 구성되어 있어야 한다.
- 마우스 스크롤시 밑에 있는 소개글 블록과 시나리오 박스가 자연스럽게 올라와야 한다.

## 2.0 Generative AI 기능

## 2.1 생성형 AI 대화 기능

- 사용자가 질문을 입력한 후 엔터키 또는 전송 버튼을 클릭하면 로컬에서 **Ollama** 기반 **LLaMA3** 모델은 해당 질문에 실시간으로 답변을 생성하여 출력 할 수 있어야 한다.
- 질문에 대한 정보는 꼭 **RAG**에서 검색 후에 컨텍스트 생성 후 답변을 생성해야 한다.
- 답변은 무조건 한국어로 제공되어야 한다.
- 참고 문헌의 내용이 다른나라 언어더라도 번역 후 한국어로 답변해주어야하며 사용자에게 문헌을 보여줄때에도 번역해서 보여줄 수 있도록 한다.
- 사용자와의 대화는 세션을 통해서 저장할 수 있도록 한다.
- 답변이 길어질 경우 스크롤로 전체 내용을 확인할 수 있어야 한다.
- 사용자 입력과 모델 응답은 말풍선 **UI**로 시간순으로 표시되어야 한다.
- 답변 생성 시에는 로딩 상태를 시각적으로 봇 아이콘이 깜빡이며 진행중임을 인지할 수 있도록 표시 되어야 한다.
- 새로운 메시지가 도착하면 자동으로 하단으로 스크롤 이동해야 한다.
- “대화 초기화” 버튼 클릭 시 대화 내용이 전부 초기화 되어야 한다.
- 모델 응답은 최대 **1000**자 이내로 제한되며, 불필요한 답변은 최소화되어야 한다.

## 2.2 텍스트 요약 기능

- 요약에는 **csebuetnlp/mT5\_multilingual\_XLSum** 모델을 사용해야 한다.
- 사용자가 긴 문단을 입력하면 모델이 주요 내용을 추출하여 한국어로 요약해야 한다. (단, 요청시에만 요약을 할 수 있도록 한다.)
- 요약은 **3~5**줄 정도의 간결한 문장으로 출력되어야 한다.
- 요약 결과는 대화창에 메시지 형태로 출력되어야 한다.
- 불필요한 세부 사항은 제거하고 핵심 맥락만 남겨야 한다.
- 입력 길이 제한을 두어 모델이 안정적으로 처리할 수 있어야 한다.

## 2.3 번역 기능

- 사용자가 한국어 텍스트를 입력하면 영어로, 영어 텍스트를 입력하면 한국어로 번역할 수 있어야 한다. (단, 요청시에만 번역을 할 수 있도록 한다.)

- 번역 결과는 대화창에 메시지 형태로 출력되어야 한다.
- 번역 품질은 문맥을 고려해 자연스럽게 생성되어야 한다.
- 번역 모델은 **facebook/nllb-200-distilled-600M** 을 사용해야 한다.
- 긴 문장의 경우 입력 제한을 두고, 오류 발생 시 사용자에게 안내 메시지를 제공해야 한다.

### 3.0 RAG 기반 산불 정보 응답 기능

#### 3.1 문서 임베딩 및 인덱싱

- 문서 임베딩은 **BAAI/bge-m3** 모델을 사용해 클라우드에서 수행해야 한다.
- 벡터데이터베이스는 **Qdrant**를 사용해야 한다.
- 참고할 문서들은 산불관련 문서, 과거 산불데이터, 정책 문서등으로 **PDF**로 받아와서 넣도록 한다.
- 임베딩된 벡터는 **Qdrant**에 저장되어야 한다.
- 저장 시 벡터와 함께 문헌 **ID**, 문서명 등의 메타데이터가 **Qdrant**에 기록되어야 한다.

#### 3.2 벡터 검색

- 사용자가 산불 관련 질문을 입력하면 질문을 벡터화하여 **Qdrant**에서 검색을 수행해야 한다.
- 검색은 코사인 유사도를 기준으로 해야한다.

#### 3.3 재순위화(리랭크)

- 재순위화 하는 모델은 **BAAI/bge-reranker-large** 모델을 사용해야 한다.
- 검색된 후보 문서는 클라우드에서 재순위화된다.
- 재순위화 결과는 적합도가 높은 순으로 정렬되어 **LLM**에 전달되어야 한다.

#### 3.4 응답 생성

- 최종 응답은 클라우드에서 **Meta LLaMA3**가 수행하도록 한다.
- **LLaMA3**는 재순위화된 문헌을 컨텍스트로 사용하여 답변을 생성해야 한다.

- 답변은 무조건 한국어로 제공되어야 하며, 1000자 이내로 요약할 수 있도록 한다.
- 답변 하단에는 참고 문헌이 각주 형태로 인용되어야 하며 정확도 점수가 높은 순서로 나열되어야 한다.
- 참고 문헌의 내용이 다른나라 언어더라도 번역 후 한국어로 답변해주어야하며 사용자에게 문헌을 보여줄때에도 번역해서 보여줄 수 있도록 한다.

### 3.5 예외 처리

- 검색 결과가 없을 경우 “관련 문헌 없음” 메시지를 표시해야 한다.
- API 호출 지연/오류 발생 시 사용자에게 오류 메시지를 제공해야 한다.

## 4.0 Vision (YOLO 기반 산불 조기 감지)

### 4.1 동영상 업로드

- 사용자는 동영상을 업로드하여 산불 탐지를 요청할 수 있어야 한다.
- 업로드 후 미리보기가 제공되어야 한다.

### 4.2 탐지 및 결과 표시

- YOLO 모델은 불과 연기를 탐지해야 한다.
- 탐지된 객체는 영상 위에 바운딩 박스 및 클래스 레이블로 표시되어야 한다.
- 결과는 비디오 오버레이 형태로 사용자에게 시각적으로 제공되어야 한다.
- 이벤트 발생 시간과 탐지 개수는 요약 카드 형태로 함께 표시되어야 한다.

## 5.0 ML 기반 산불 확산 예측 기능

### 5.1 입력 기능

- 사용자는 시/군을 드롭다운으로 선택해야 한다.
- 지역은 Oracle DB에서 관리되며, 선택 시 해당 지역의 기준 좌표가 자동 입력된다.
- 사용자는 지도 클릭으로 위도/경도를 직접 수정할 수도 있어야 한다.
- 예측 시점(3시간 후, 6시간 후, 9시간 후)을 드롭다운으로 선택할 수 있어야 한다.

## 5.2 예측 실행

- 지도를 클릭하여 위경도를 선택한 후 예측 실행 버튼 클릭 시 백엔드 **API**가 호출되어 실시간으로 **NASA POWER**의 기상 데이터와 **Google Earth Engine**에서 지형 데이터와 위성 데이터를 불러와 모델이 계산 할 수 있도록 한다
- 예측 실행 버튼 클릭 시 백엔드 **API**가 호출되어 머신러닝 모델이 결과를 생성해야 한다.
- 요청 파라미터에는 지역명, 좌표, 예측 시점이 반드시 포함되어야 한다.

## 5.3 출력 기능

- 예측 결과는 **Kakao Maps** 위에 팝업으로 표시되어야 한다.
- 출력 지표는 피해 면적(**ha**), 확산 방향, 확산 거리(**m**), 확산 속도(**m/h**)로 표시될 수 있어야한다.
- 속도는 빠름 / 보통 / 느림 세 단계 라벨로 변환되어야 한다.
- 피해면적과 확산 방향, 확산 거리, 확산 속도는 지도에서 결과가 나오면 사용자가 보기 쉽게 아래에도 표시가 되어야 한다

## 5.4 예외 처리

- 예측 실행 시 지연이 발생하면 로딩 인디케이터를 표시해야 한다.
- 오류가 발생할 경우 사용자에게 메시지를 제공할 수 있어야 한다.