

개인 맞춤형 알레르기 정보 기반의 스마트 식품 필터 개발에 관한 연구

곽건(국립군산대학교), 조주필 교수(국립군산대학교)

rhkrrjs0221@naver.com, stefano@kunsan.ac.kr, kc.kang@kunsan.ac.kr

A Study on the Development of a Smart Food Filter based on Personalized Allergy Information

Geon Gwak, Jupil Jo

요약

본 논문은 개인의 알레르기 정보에 기반하여 식품의 유해 성분을 신속하게 판별하는 웹 애플리케이션 '스마트 알러지 식품 필터' 개발에 관한 연구이다. 현대 사회의 가공식품 소비 증가에 따라 식품 알레르기 문제가 중요한 건강 위협 요소로 대두되고 있으나, 소비자가 매번 식품 성분을 직접 확인하는 것은 매우 번거롭다. 이러한 문제를 해결하기 위해 Python의 Flask 프레임워크를 기반으로 백엔드 서버를 구축하고, 로컬 데이터베이스와 공공 API를 연동하여 실시간으로 식품 성분을 분석하는 시스템을 구현하였다. 특히, 사용자의 알레르기 정보와 식품 성분 정보를 Python의 집합(Set) 자료구조를 활용하여 효율적으로 비교하고, 위험 성분 포함 여부를 즉각적으로 사용자에게 경고하는 핵심 알고리즘을 개발하였다. 본 논문에서는 시스템의 전체적인 구조와 기능 구현 과정, 그리고 외부 API 연동 시 발생한 SSL 보안 프로토콜 충돌 문제와 그 해결 방안에 대해 상세히 다루며, ICT 기술을 활용한 생활 밀착형 건강 관리 서비스의 개발 성과와 기대효과를 논의한다.

I. 서론

현대 사회에서 가공식품의 소비가 보편화되면서 식품 알레르기는 개인의 건강을 위협하는 중요한 문제로 자리 잡았다. 특히 면역 체계가 민감한 특정 계층에게 소량의 알레르기 유발 성분은 심각한 건강 문제로 이어질 수 있다. 그러나 대부분의 식품 성분 정보는 포장지에 작은 글씨로 표기되어 있거나 온라인 정보가 부족하여, 소비자가 알레르기 유발 성분을 매번 확인하는 것은 매우 어려운 일이다.

본 논문은 이러한 문제를 해결하기 위해 사용자의 개인별 알레르기 정보를 바탕으로 식품의 안전성을 신속하게 판단해주는 '스마트 알러지 식품 필터' 웹 애플리케이션 개발에 대한 연구를 다룬다. 이를 통해 정보통신기술(ICT)을 활용하여 사용자가 일상에서 더욱 편리하고 안전하게 식품을 선택할 수 있는 생활 밀착형 건강 관리 서비스의 기반을 마련하고자 한다. 본 논문에서는 다음과 같은 내용들을 다룬다.

1. 시스템 설계 및 구현: Python Flask 프레임워크를 이용한 백엔드 서버와 프론트엔드 구성, 그리고 사용자별 알레르기 정보 관리 기능에 대한 설명.
2. 핵심 알고리즘: 로컬 DB 및 외부 공공 API를 연동하여 식품 성분을 실시간으로 분석하고, 사용자의 프로필과 비교하여 위험도를 판단하는 알고리즘에 대한 설명.
3. 문제 해결 과정: 외부 API 연동 시 발생한 기술적 문제점(SSL 보안 프로토콜 충돌)과 이를 해결한 방안에 대한 상세한 기술.

마지막으로 개발 결과를 정리하고, 본 연구의 성과와 향후 발전 가능성에 대해 논하며 결론을 맺는다.

II. 본론

1. 시스템 구성 및 개발 환경

본 시스템은 Python의 Flask 프레임워크를 기반으로 한 백엔드 서버와 HTML/CSS/JavaScript로 구성된 프론트엔드로 개발되었다. 백엔드 서버는 RESTful API 형식을 따르며, 사용자의 알레르기 정보 관리, 식품 정보 조회 및 위험성 판단의 핵심 기능을 수행한다. 데이터는 로컬 메모리에 저장된 딕셔너리 형태의 데이터베이스와 외부 '식품안전나라'의 공공 API를 함께 사용하여 조회 속도와 정보의 정확성을 모두 확보하고자 하였다.

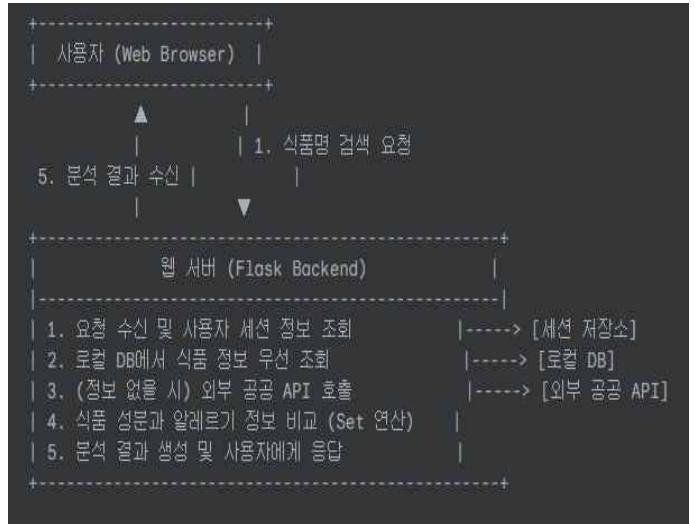


그림 1. 시스템 구성도

2. 주요 기능 및 알고리즘 구현

웹 애플리케이션의 핵심 기능은 '데이터 조회 → 사용자 정보 연동 → 위험성 판단'의 3단계 알고리즘으로 구현되었다.

1. 개인화된 알레르기 정보 관리

아래 그림 2.는 개인화 알레르기 정보 관리의 구조를 나타낸다.

사용자는 회원가입이나 프로필 설정을 통해 자신의 알레르기 유발 물질 정보를 시스템에 등록할 수 있다. 서버는 Flask의 session을 이용해 각 사용자를 고유 ID로 식별하고, 서버 메모리에 사용자의 프로필(알레르기 정보)을 저장하고 관리하는 API(/api/profile)를 구현하였다.

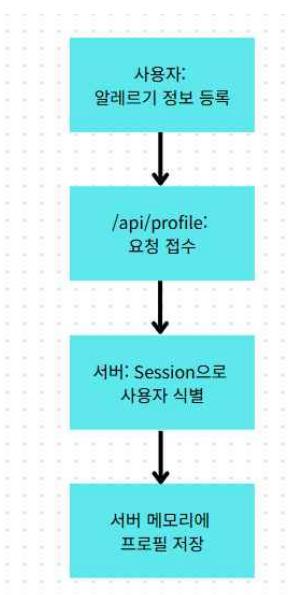


그림 2. 개인화 알레르기 정보 관리

2. 실시간 식품 성분 분석 및 위험성 판단

아래 그림 3은 실시간 식품 성분 분석 및 위험성 판단구조를 나타낸다. 사용자가 식품명을 검색하면, 백엔드 API(/api/search_food)가 호출된다. 이 API의 동작 과정은 다음과 같다.

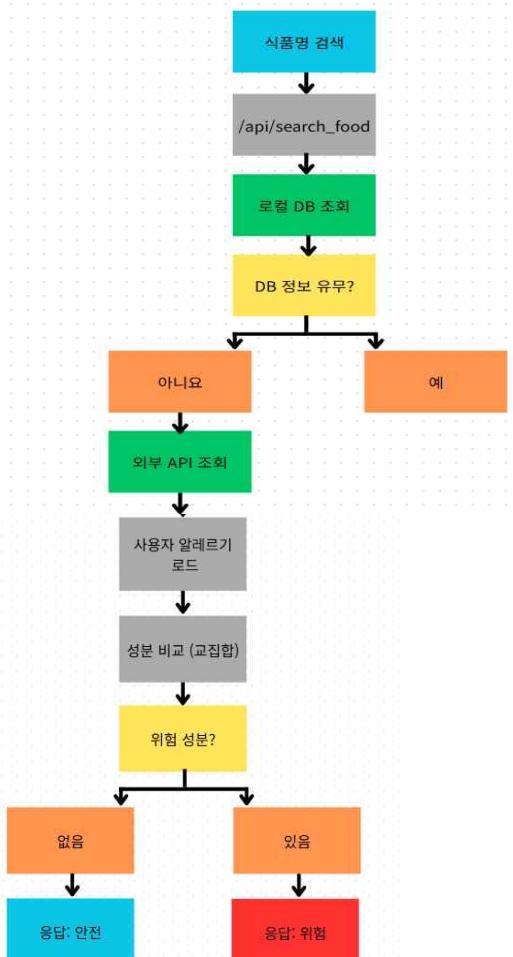


그림 3. 실시간 식품 성분 분석 및 위험성 판단

- 클라이언트로부터 food_name(식품명)을 전달받는다.
- 로컬 DB에서 해당 식품 정보를 우선 조회하여 응답 속도를 높인다.
- 로컬 DB에 정보가 없을 경우, 외부 공공데이터포털 API를 호출하여 성분 정보를 조회한다.
- 현재 session에 저장된 사용자의 알레르기 정보(user_allergies)를 가져온다.
- Python의 set(집합) 자료구조를 활용하여, 조회된 식품의 알레르기 성분(food_allergens)과 사용자 알레르기 정보의 **교집합(intersection)**을 계산하여 겹치는 위험 성분(matching_allergens)을 찾아낸다.
- 분석 결과(겹치는 성분 유무, 정보 유무 등)에 따라 reaction_found, safe, no_info 등의 상태 값을 포함한 JSON 형식으로 클라이언트에 응답한다.

이 알고리즘은 집합 연산을 통해 다수의 알레르기 정보를 매우 효율적으로 비교할 수 있게 한다.

3. 과제 수행 시 문제점 및 해결방안

1. 문제점: 외부 공공 API 접속 불가 (SSL 보안 프로토콜 충돌)

개발 과정에서 공공데이터포털 API 서버의 보안 정책(SSL/TLS)과 Python requests 라이브러리의 기본 암호화 방식이 호환되지 않아 SSLError가 발생하는 문제가 있었다. 이로 인해 API 호출이 원천적으로 차단되어 데이터 조회가 불가능했다.

2. 해결 방안

단순히 SSL 인증을 비활성화하는 verify=False 옵션으로는 해결되지 않는 문제임을 파악하고, 서버와 클라이언트 간의 암호화 규칙(Cipher Suite)을 강제로 지정하는 방법을 도입했다. requests.adapters.HTTPAdapter를 상속받는 TlsAdapter 클래스를 직접 구현하여, 서버가 요구하는 보안 수준(DEFAULT@SECLEVEL=1)에 맞는 SSL 컨텍스트를 생성하도록 하였다. 이 커스텀 어댑터를 requests.Session 객체에 마운트하여 API를 호출함으로써 보안 프로토콜 충돌 문제를 해결하고 안정적인 통신에 성공하였다.

III. 결론

본 연구에서는 개인 맞춤형 알레르기 정보를 기반으로 식품의 안전성을 실시간으로 필터링하는 웹 애플리케이션을 성공적으로 개발하였다. 사용자는 식품명을 검색하는 간단한 행위만으로 잠재적인 알레르기 위험을 즉시 확인할 수 있으며, 시스템은 로컬 DB와 외부 공공 API를 유기적으로 연동하여 빠르고 정확한 정보를 제공한다.

본 연구의 기술적 성과는 Python Flask 기반의 웹 백엔드 서버 설계 역량을 확보하고, 외부 API 연동 과정에서 발생한 SSL 보안 문제를 직접 분석하여 해결하는 실무적인 능력을 증명했다는 점에 있다. 또한, 세션 기반의 사용자 관리와 데이터 집합(Set)을 활용한 효율적인 데이터 비교 알고리즘을 구현하였다.

기대효과로는 식품 알레르기로 불편을 겪는 사람들이 더 안전하고 편리하게 식품을 소비할 수 있도록 돋는 실용적인 서비스를 제공하여 사회적 가치를 창출할 수 있다는 점을 들 수 있다. 향후 이미지 인식(OCR) 기술을 접목하여 제품 포장지를 촬영하면 자동으로 성분을 분석해주는 기능으로 발전시킬 수 있을 것이다. 본 연구는 공공 데이터를 활용한 생활 밀착형 ICT 서비스 개발의 좋은 사례로서, 향후 유사 프로젝트에 중요한 참고 자료가 될 것으로 기대한다.

ACKNOWLEDGMENT

본 논문은 (재)전북테크노파크 재원을 지원받아 수행된 지역특성화산업 전문인력양성사업 연구 결과입니다.

참 고 문 헌

- [1] Mark Pilgrim, "Dive Into Python 3," Apress, 2009.
- [2] Miguel Grinberg, "Flask Web Development: Developing Web Applications with Python," O'Reilly Media, 2018.
- [3] 식품의약품안전처, "식품안전나라 식품영양성분DB API 활용가이드," 2023.
- [4] J. D. Saltzer, "The Protection of Information in Computer Systems," Proceedings of the IEEE, vol. 63, no. 9, pp. 1278-1308, 1975.