

# 웹 크롤링 기반 개인 맞춤형 식품 알레르기 정보 시스템



21012047 인공지능학과 강민성

제출일 : 2025년 6월 22일

# 목차

## 1. 서론

1.1. 프로젝트 배경 및 필요성

1.2. 프로젝트 목표 및 범위

## 2. 데이터베이스 설계

2.1. 요구사항 분석

2.2. ERD 설계

2.3. 논리적 설계

2.4. 물리적 설계

## 3. 시스템 구현

3.1. 개발 환경

3.2. 데이터 수집 및 적재

3.3. 핵심 기능 구현

## 4. 실행 결과

## 5. 결론

5.1. 프로젝트 요약 및 성과

5.2. 고찰 및 향후 개선 방안

# 1. 서론

## 1.1. 프로젝트 배경 및 필요성

현대 사회에서 식품 알레르기는 더 이상 소수만의 문제가 아닌, 모든 소비자에게 중요한 건강 및 안전 이슈로 자리 잡았다. 식품의약품안전처는 알레르기 유발 물질 표시 대상을 지속적으로 확대하고 있으며, 식품 제조사들은 관련 정보를 자사 제품에 의무적으로 표기하고 있다. 하지만 소비자 입장에서 특정 제품의 알레르기 정보를 얻기 위해서는 개별 제조사의 웹사이트를 방문하거나 제품의 포장지를 직접 확인해야 하는 불편함이 따른다. 정보가 여러 채널에 분산되어 있어 통합적인 정보에 대한 접근성이 현저히 낮다.

본 프로젝트의 직접적인 계기는 미디어에서 한 등장인물이 음식에 자신에게 알레르기를 유발하는 물질이 들어있는지 모르고 음식을 섭취한 후, 심각한 알레르기 반응으로 응급 상황에 처하는 장면을 접한 것에서 시작되었다. 이처럼 정보의 부재는 개인의 안전에 심각한 위협이 될 수 있다는 문제의식을 갖게 되었다. 이에, 식품별 알레르기 정보를 한곳에서 쉽게 확인할 수 있는 서비스의 필요성을 느끼고 관련 시장을 조사하였다.

조사 결과, 스마트폰 카메라를 이용해 제품의 바코드나 이미지를 스캔하여 정보를 제공하는 애플리케이션은 소수 존재했다. 하지만 이러한 방식은 사용자가 실제 제품을 소지하고 있어야만 한다는 문제점이 있다. 반면, 웹 크롤링을 통해 최신 제품 정보를 지속적으로 수집하고, 사용자가 제품명으로 직접 검색하여 관련 알레르기 정보, 대체 식품, 나아가 교차 반응 정보까지 종합적으로 제공하는 방식의 서비스는 존재하지 않는다는 것을 확인하였다.

따라서, 분산된 데이터를 통합하고 사용자 중심의 검색 환경을 제공하여 식품 알레르기 정보의 접근성을 획기적으로 개선할 수 있는 시스템의 필요성은 매우 높다고 할 수 있다. 본 프로젝트는 이러한 필요성에 따라, 사용자의 안전한 식품 선택을 돕는 통합 데이터베이스 및 응용 프로그램 개발을 목표로 한다.

## 1.2. 프로젝트 목표 및 범위

본 프로젝트는 앞서 서술한 배경 및 필요성을 바탕으로, 다음과 같은 구체적인 목표를 설정하고 그에 따른 범위를 정의하였다.

- 첫째, 여러 식품 제조사 웹사이트에 흩어져 있는 제품 및 알레르기 정보를 웹 크롤링으로 수집하고, 이를 효율적으로 관리할 수 있는 정규화된 관계형 데이터베이스를 설계 및 구축한다.
- 둘째, 사용자가 제품명으로 검색 시, 단순 알레르기 정보뿐만 아니라 관련 대체 식품과 교차반응군 정보까지 종합적으로 제공하는 python 기반 애플리케이션을 개발한다.
- 셋째, 사용자 계정 시스템을 도입하여 비밀번호 해싱을 통한 안전한 로그인 환경을 제공하고, 사용자가 등록한 정보에 기반한 개인 맞춤형 알레르기 경고 기능을 구현한다.
- 넷째, 시스템 데이터 현황을 한눈에 파악할 수 있도록 데이터 분석 기능이 포함된 관리자 전용 메뉴를 구현한다.

본 프로젝트에서 구현한 기능의 범위는 다음과 같다.

- **데이터베이스:**

Company, Food, Allergy, Alternative\_Food, Cross\_Reaction, Users 등 10개의 테이블로 구성된 스키마 설계 및 구축

데이터 무결성을 위한 PK, FK, Unique, Not Null, Default 제약조건 적용

참조 무결성을 위한 ON DELETE CASCADE 옵션 적용

- **핵심 기능:**

제품명 기반 식품의 종합 알레르기 정보(대체 식품, 교차 반응군 포함) 조회

- **사용자 기능:**

보안을 고려한 사용자 회원가입 및 로그인, 로그아웃

개인 알레르기 정보 등록 및 관리

비밀번호 확인을 통한 안전한 회원 탈퇴

- **관리자 기능:**

역할 기반 관리자 접근 제어

전체 사용자 목록 조회

사용자 등록 기반 알레르기 순위 통계 분석(TOP 5)

- **데이터 관리 기능:**

신규 제품, 알레르기 정보의 추가, 수정, 삭제(CRUD)

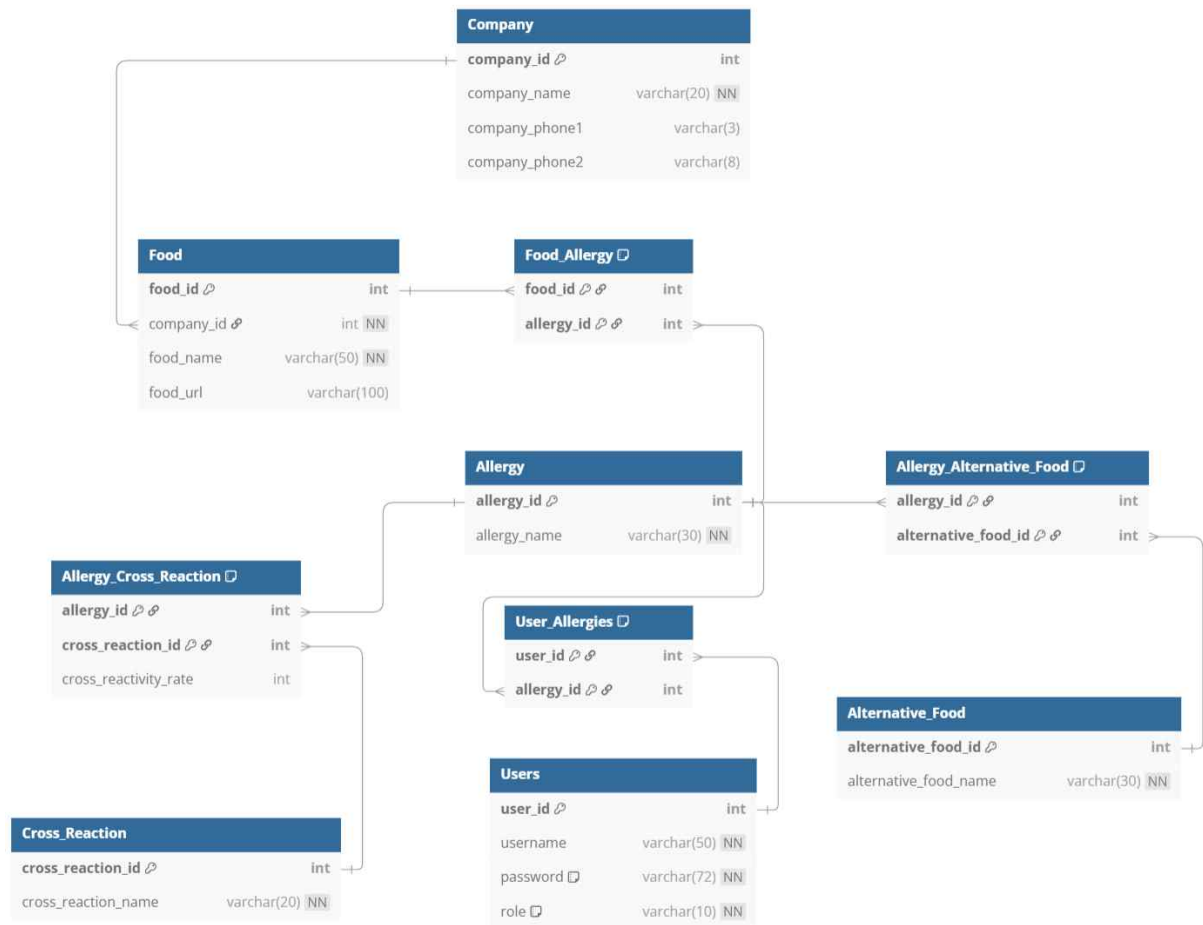
## 2. 데이터베이스 설계

### 2.1. 요구사항 분석

- 사용자는 시스템에 새로운 계정을 생성(회원가입)할 수 있어야 한다.
- 사용자는 아이디와 비밀번호를 통해 시스템에 접근(로그인)할 수 있어야 한다.
- 로그인한 사용자는 자신의 계정을 영구적으로 삭제할 수 있어야 한다.
- 로그인한 사용자는 자신의 알레르기 정보를 시스템에 등록하고 관리할 수 있어야 한다.
- 시스템은 '일반 사용자'와 '관리자'의 역할을 구분하여 관리할 수 있어야 한다.
- 시스템은 식품, 제조사, 알레르기, 대체 식품, 교차 반응군에 대한 정보를 저장하고 관리할 수 있어야 한다.
- 모든 사용자는 제품명을 통해 식품 정보를 검색할 수 있어야 한다.
- 검색 결과에는 해당 식품의 알레르기 유발 물질, 대체 식품, 교차 반응군 정보가 종합적으로 포함되어야 한다.
- 로그인한 사용자가 식품을 검색할 경우, 해당 식품에 자신의 알레르기 유발 물질이 포함되어 있는지 여부를 개인 맞춤형 알림으로 제공해야 한다.
- 관리자는 시스템에 등록된 전체 사용자 목록을 조회할 수 있어야 한다.
- 관리자는 사용자들이 등록한 알레르기 정보에 대한 통계를 조회할 수 있어야 한다.
- 사용자의 비밀번호는 평문으로 저장되어서는 안되며, 반드시 해싱하여 저장해야 한다.
- 각 데이터 테이블 간의 관계는 명확해야 하며, 특정 데이터가 삭제될 때 연관된 데이터가 유실되거나 고아 데이터가 남지 않도록 제약조건을 설정해야 한다.
- 사용자는 별도의 학습 없이도 기능을 쉽게 이해할 수 있도록, 명확한 텍스트 기반의 메뉴 인터페이스를 제공해야 한다.

## 2.2. ERD 설계

앞서 정의한 요구사항을 만족시키기 위해 데이터베이스의 전체적인 구조를 나타내는 ERD 설계를 진행하였다. 시스템의 핵심 데이터 요소들을 개체로 정의하고 개체 간의 관계를 모델링하여 아래와 같은 최종 ERD를 도출하였다.



- Users: 사용자 정보를 저장하는 개체. 아이디, 해시된 비밀번호, 역할 정보를 포함한다.
- Company: 오투기, CJ제일제당 등 식품 제조사 정보를 저장한다.
- Food: 각 회사에서 생산하는 식품의 이름, 정보 URL 등을 저장한다.
- Allergy: 난류, 우유, 대두 등 알레르기 유발 물질의 종류를 정의한다.
- Alternative\_Food: 특정 알레르기 성분을 대체할 수 있는 식품 정보를 저장한다.
- Cross\_Reaction: 특정 알레르기 성분과 교차 반응을 일으킬 수 있는 식품군 정보를 저장한다.

본 데이터베이스의 핵심은 여러 개체 간의 다대다 관계를 어떻게 효율적으로 설계하는지에 있었다. 하나의 식품은 여러 알레르기 성분을 가질 수 있고, 하나의 알레르기 성분은 여러 식품에 포함될 수 있다. 이 다대다 관계를 해소하기 위해 Food\_Allergy라는 연결 테이블을 설계하였다.

마찬가지로, Allergy와 Alternative\_Food, Allergy와 Cross\_Reaction간의 관계 역시 다대다 관계이므로, 각각 Allergy\_Alternative\_Food와 Allergy\_Cross\_Reaction 연결 테이블을 만들어 정규화를 만족시켰다.

사용자 개인화 기능을 위해 한 명의 사용자가 여러 알레르기 정보를 가질 수 있는 다대다 관계를 User\_Allergies 연결 테이블로 설계하였다.

이러한 설계를 통해 데이터의 중복을 방지하고, 각 테이블이 하나의 주제에만 집중하도록 하여 데이터베이스의 유연성과 확장성을 확보하였다.

## 2.3. 논리적 설계

ERD 설계를 통해 도출된 ERD를 관계형 데이터베이스 모델에 맞춰 변환하는 논리적 설계를 진행하였다. 이 과정에서 테이블, 컬럼, 제약조건 등을 구체적으로 정의하여 데이터의 일관성과 무결성을 확보하고자 하였다

### ● 테이블 및 컬럼 정의:

개념적 설계의 각 개체는 데이터베이스의 테이블로, 각 속성은 테이블의 컬럼으로 매핑하였다.

최종적으로 Company, Food, Allergy, Alternative\_Food, Cross\_Reaction, Users, User\_Allergies 등 총 10개의 테이블을 정의하였다.

각 컬럼에는 INT, VARCHAR 등 데이터의 특성과 예상 길이를 고려하여 최적의 데이터 타입을 지정하였다.

### ● 기본키(Primary Key) 및 고유 제약 조건:

Users, Food 등 독립적인 개체 테이블은 AUTO\_INCREMENT 속성을 가진 정수형 id 컬럼을 단일 기본키로 설정하여 데이터의 유일성을 보장하였다.

Food\_Allergy, User\_Allergies 등 다대다 관계를 위한 연결 테이블은, 두 부모 테이블의 외래키를 조합한 복합키를 기본키로 설정하여 관계 자체의 중복을 방지하였다.

Users 테이블의 username 컬럼에는 UNIQUE 제약조건을 부여하여, 동일한 아이디가 중복으로 생성되는 것을 시스템 차원에서 방지하였다.

### ● 외래키(Foreign Key) 및 참조 무결성:

테이블 간의 논리적 관계는 외래키 제약조건을 통해 명확히 정의하였다.

특히, 모든 외래키 관계에는 ON DELETE CASCADE 옵션을 적용하였다. 예를 들어, 특정 사용자가 Users 테이블에서 삭제될 경우, User\_Allergies 테이블에 저장된 해당 사용자의 모든 알레르기 정보가 자동으로 삭제된다. 이를 통해 데이터의 참조 무결성을 유지하고 고아 데이터 발생을 방지하였다.

- 정규화:

위와 같은 설계를 통해 데이터의 중복을 제거하고, 각 테이블이 하나의 주제에만 집중하도록 하여 데이터베이스의 제 3정규형을 만족시켰다. 이는 데이터 수정 시 발생할 수 있는 이상 현상을 최소화하고 데이터의 일관성을 유지하는 데 기여한다.

## 2.4. 물리적 설계

논리적 설계를 통해 정의된 관계형 스키마를 특정 데이터베이스 관리 시스템에 실제로 구축하는 물리적 설계 단계를 진행하였다. 이 단계에서는 실제 데이터 저장 방식과 시스템의 성능까지 고려하여 세부 사항을 결정하였다.

- DBMS 선정:

본 프로젝트에서는 널리 사용되며 안정성이 검증된 오픈소스 RDBMS인 MySQL 8.0을 최종 시스템으로 채택하여 데이터베이스를 구축하였다.

- 비밀번호 저장을 위한 컬럼 설계:

사용자의 비밀번호는 보안을 위해 bcrypt 해싱 알고리즘을 사용하여 암호화하였다. bcrypt 해시값은 항상 60자의 고정된 길이를 가지므로, Users 테이블의 password 컬럼 데이터 타입을 VARCHAR(72)로 넉넉하게 설정하여 데이터의 잘림 없이 안전하게 저장될 수 있도록 하였다.

- 기본키 자동 증가:

모든 독립 개체 테이블의 기본키 컬럼에는 AUTO\_INCREMENT 속성을 부여하여, 새로운 데이터가 추가될 때마다 시스템이 자동으로 고유한 식별자를 할당하도록 구현하였다. MySQL은 기본키에 대해 자동으로 B-Tree 인덱스를 생성하므로, 테이블 간 JOIN이나 특정 데이터 조회 시 빠른 속도를 보장할 수 있다.



### 3. 시스템 구현

#### 3.1. 개발 환경

본 프로젝트의 개발 및 실행을 위해 사용된 환경은 다음과 같다.

- **DBMS:** MySQL 8.0
- **개발 언어:** Python 3.10
- **주요 라이브러리:**

pymysql: Python과 MySQL 데이터베이스 간의 연동 및 쿼리 실행

bcrypt: 사용자 비밀번호의 안전한 해싱 및 검증

pandas: CSV 파일 형식의 크롤링 데이터 처리

requests, BeautifulSoup4: 웹페이지 데이터 크롤링

## 3.2. 데이터 수집 및 적재

본 프로젝트에 필요한 데이터는 상용 서비스를 이용하지 않고, Python을 활용한 웹 크롤링을 통해 직접 수집하였다. 이 과정은 동적 웹페이지의 데이터를 수집하고, 이미지 내의 텍스트를 인식하는 등 여러 기술적 과제를 해결하는 과정으로 이루어졌다.

- 수집 대상:

‘오투기물’ 웹사이트의 전체 카테고리에 등록된 제품 약 1,152개의 제품명, 상세 페이지 URL, 그리고 상세 페이지에 이미지로 포함된 알레르기 유발 물질 정보

- 세션 관리 및 AJAX 요청:

웹사이트가 사용자의 탐색 상태를 유지하기 위해 세션을 사용하므로, `requests.Session()` 객체를 생성하여 크롤링 내내 동일한 세션을 유지하도록 구현하였다.

제품 목록은 초기 HTML 페이지에 포함되어 있지 않고, 사용자가 스크롤하거나 페이지 번호를 클릭할 때마다 비동기적으로 로드되는 AJAX 방식이었다. 이를 해결하기 위해, 브라우저의 개발자 도구로 네트워크 요청을 분석하여 실제 데이터 목록을 반환하는 `product_list.ajax` URL을 찾아내고, 이 주소로 POST 요청을 보내는 방식으로 전체 페이지의 데이터를 수집하였다.

- 이미지 내 텍스트 추출(OCR):

본 프로젝트의 가장 큰 기술적 과제는, 핵심 정보인 알레르기 유발 물질 목록이 텍스트가 아닌 이미지 파일 안에 포함되어 있다는 점이었다.

이 문제를 해결하기 위해, 각 제품의 상세 페이지에서 마지막 상세 정보 이미지를 다운로드한 후, Tesseract OCR 엔진을 Python에서 제어할 수 있는 `pytesseract` 라이브러리를 사용하여 이미지로부터 텍스트를 추출하였다.

OCR 엔진의 이미지 크기 제한으로 인한 오류를 방지하기 위해, 특정 크기를 초과하는 이미지는 하단 영역만 잘라내어 처리하는 `ocr_with_size_check` 함수를 별도로 구현하여 안정성을 높였다.

초기 테스트에서 Tesseract의 한글 인식률이 일부 저하되는 문제를 인지하고, 정확도 개선을 위해 PaddleOCR 라이브러리의 도입을 시도하였다. 하지만 개발 환경의 CUDA 버전과 cuDNN 버전의 호환성 문제가 발생하여 최종적으로는 안정성이 확보된 Tesseract OCR을 사용하는 것으로 결정하였다. 이 과정은 새로운 기술 도입 시 개발 환경의 복잡한 의존성 관리의 중요성을 배우는 계기가 되었다.

- 서버 부하 방지를 위한 예외 처리:

웹사이트 서버에 과도한 부하를 주지 않기 위해, 각 제품 페이지를 요청할 때마다 `time.sleep(random.uniform(1.0, 10.0))` 코드를 사용하여 1초에서 10초 사이의 임의의 지연 시간을 주었다.

- **데이터 정제:**

크롤링으로 수집한 제품명에는 [행사]나 (증정)과 같은 불필요한 홍보 문구가 포함되어 있었다. 정규표현식을 사용하여 이러한 패턴들을 제거하는 `clean_product_name` 함수를 구현하여 데이터의 일관성과 품질을 확보하였다.

- **알레르기 정보 식별:**

OCR을 통해 이미지에서 추출된 전체 텍스트 중에서, 미리 정의해 둔 알레르기 물질 이름(`allergy_dict`)이 포함되어 있는지 여부를 검사하여 최종 알레르기 목록을 식별하였다.

### 3.3. 핵심 기능 구현

데이터베이스 설계 파트에서 설계한 데이터베이스를 기반으로, 요구사항 분석 파트에서 정의한 요구사항을 만족시키기 위해 Python 애플리케이션의 핵심 기능들을 다음과 같이 구현하였다.

- **사용자 계정의 전체 관리 기능 구현:**

사용자가 계정을 생성하고, 시스템에 접근하며, 필요시 안전하게 탈퇴할 수 있는 기능들을 `main.py`에 `register_func`, `login_func`, `delete_account_func` 함수로 각각 구현하였다.

- **비밀번호 해싱 적용:**

사용자의 비밀번호는 `bcrypt` 라이브러리를 사용하여 암호화된 해시값으로 변환한 후 `Users` 테이블에 저장된다. 로그인 및 회원 탈퇴 시에는 사용자가 입력한 비밀번호를 동일한 방식으로 해싱하여 DB에 저장된 값과 비교하는 `checkpw` 함수를 통해 인증 과정을 안전하게 처리하였다.

- **데이터 무결성 보장:**

`Users` 테이블의 사용자가 삭제될 때, `ON DELETE CASCADE` 제약조건에 따라 `USER_Allergies` 테이블에 저장된 해당 사용자의 모든 알레르기 정보가 자동으로 함께 삭제되도록 하며, 데이터의 일관성과 무결성을 시스템 차원에서 유지하였다.

- **개인 알레르기 정보 관리:**

로그인한 사용자는 `my_allergy_func` 메뉴를 통해 자신의 알레르기 정보를 `User_Allergies` 테이블에 등록하고 조회할 수 있다.

- **맞춤형 알림 로직:**

식품 조회시, 시스템은 현재 로그인된 사용자가 있는지 확인한다. 로그인 상태일

경우, 해당 식품의 알레르기 정보와 사용자가 등록한 알레르기 정보를 비교하여, 일치하는 항목이 있을 시 개인화된 경고 메시지를 추가로 출력하도록 구현하였다.

- 권한 분리:

Users 테이블에 role 컬럼을 두어 사용자와 관리자 권한을 분리하였다. main.py에서는 로그인된 사용자의 role 값을 확인하여, 'admin'일 경우에만 관리자 전용 메뉴가 노출되도록 구현하였다.

- 사용자 관리 및 데이터 분석:

관리자는 전용 메뉴를 통해 시스템의 전체 사용자 목록을 조회할 수 있다. 또한 User\_Allergies와 Allergy 테이블을 JOIN하고, GROUP BY와 COUNT 집계 함수를 사용하여 '가장 많이 등록된 알레르기 순위'와 같은 유의미한 통계 정보를 동적으로 생성 및 조회할 수 있다.

- 종합 정보 조회:

식품 검색 시 JOIN 쿼리를 통해 Food, Allergy, Alternative\_Food, Cross\_Reaction 등 여러 테이블에 나뉘어 저장된 정보를 종합하여 사용자에게 한번에 제공한다.

- 데이터 CRUD:

관리자 및 일반 사용자는 제품 정보 및 알레르기 정보를 추가, 수정, 삭제할 수 있는 인터페이스를 제공받는다.

## 4. 실행 결과

3장에서 설명한 핵심 기능들이 실제로 어떻게 동작하는지 주요 시나리오에 따라 실행한 결과는 다음과 같다.

- 회원가입:

프로그램을 실행하고 '(6) 회원가입' 메뉴를 선택하여 새로운 사용자 계정을 생성한다.

아이디, 비밀번호, 그리고 비밀번호 확인을 순서대로 입력한다. 비밀번호의 유효성(8자리 이상, 일치 여부)을 검사하며, 모든 조건을 만족하면 성공적으로 계정이 생성된다는 메시지가 출력된다.

```
선택하세요 ==> 6
```

```
=== (6) 회원가입 ===
```

```
사용할 아이디: test
```

```
사용할 비밀번호: test
```

```
비밀번호 확인: test
```

```
오류: 비밀번호는 8자리 이상이어야 합니다.
```

```
선택하세요 ==> 6
```

```
=== (6) 회원가입 ===
```

```
사용할 아이디: test
```

```
사용할 비밀번호: testtest
```

```
비밀번호 확인: test1234
```

```
오류: 비밀번호가 다릅니다.
```

```
선택하세요 ==> 6
```

```
=== (6) 회원가입 ===
```

```
사용할 아이디: test
```

```
사용할 비밀번호: testtest
```

```
비밀번호 확인: testtest
```

```
'test' 계정이 성공적으로 생성되었습니다.
```

- 로그인:

메인 메뉴에서 '(7) 로그인'을 선택하고, 방금 생성한 계정의 아이디와 비밀번호를 입력한다. 로그인에 성공하면, 메뉴 상단에 "OOO님, 환영합니다!" 라는 문구가 출력되며, 사용자 전용 메뉴(내 정보 관리, 로그아웃, 회원 탈퇴)가 나타난다.

```
선택하세요 ==> 7
```

```
=== (7) 로그인 ===
```

```
아이디: test
```

```
비밀번호: testtest
```

```
'test'님, 환영합니다!
```

```
===== 식품 알레르기 확인 프로그램 =====  
(test 님, 환영합니다!)
```

```
--- 식품 정보 ---
```

- (1) 식품 알레르기 조회
- (2) 제품 정보 추가
- (3) 식품 알레르기 정보 추가
- (4) 식품 알레르기 정보 수정
- (5) 제품 정보 삭제

```
--- 사용자 메뉴 ---
```

- (6) 내 알레르기 정보 관리
- (7) 로그아웃
- (8) 회원 탈퇴

```
(0) 프로그램 종료
```

```
-----  
선택하세요 ==> █
```

● 개인 맞춤형 알림 기능 실행:

'(1) 식품 알레르기 조회' 메뉴를 선택하여 로그아웃 상태에서 '진라면 매운맛'을 검색하면, 제품의 일반적인 알레르기 정보가 출력된다.

이어서, 로그인한 사용자(예: '우유' 알레르기 등록)가 동일하게 '진라면 매운맛'을 검색하면, 일반 정보 출력 후 맨 아래에 해당 사용자를 위한 '개인 맞춤 알림' 섹션이 추가로 나타난다. 시스템은 사용자가 등록한 '우유' 정보와 제품의 '우유' 포함 정보를 비교하여 "주의! 000님께서 등록하신 '우유' 성분이 포함되어 있습니다!" 와 같은 경고 메시지를 성공적으로 출력한다.

선택하세요 ==> 1

=== (1) 식품 알레르기 유발물질 확인 ===

조회할 제품명을 입력하세요 (종료하려면 엔터): 진라면 매운맛  
조회할 회사명을 입력하세요: 오투기

'진라면 매운맛'의 알레르기 유발 성분을 조회합니다...

- 성분: 계란  
대체 식품: 두부, 콩나물, 육류  
교차반응군: 정보 없음
- 성분: 쇠고기  
대체 식품: 계란, 두부, 쇠고기, 흰 살 생선, 닭고기  
교차반응군: 정보 없음
- 성분: 돼지고기  
대체 식품: 정보 없음  
교차반응군: 정보 없음
- 성분: 닭고기  
대체 식품: 계란, 두부, 쇠고기, 돼지고기, 닭고기  
교차반응군:
  - 그 외 갑각류 (반응률: 75%)
  - 꽃게 (반응률: 75%)
  - 바닷가재 (반응률: 75%)
- 성분: 새우  
대체 식품: 계란, 두부, 쇠고기, 돼지고기, 닭고기  
교차반응군:
  - 그 외 갑각류 (반응률: 75%)
- 성분: 게  
대체 식품: 정보 없음  
교차반응군:
  - 바닷가재 (반응률: 75%)
  - 새우 (반응률: 75%)
- 성분: 오징어  
대체 식품: 계란, 두부, 쇠고기, 닭고기  
교차반응군: 정보 없음
- 성분: 고등어  
대체 식품: 정보 없음  
교차반응군: 정보 없음
- 성분: 조개류  
대체 식품: 정보 없음  
교차반응군: 정보 없음
- 성분: 전복  
대체 식품: 정보 없음  
교차반응군: 정보 없음

- 성분: 우유  
대체 식품: 깨, 호두, 잣, 아몬드  
교차반응군:
  - 그 외 콩류 (반응률: 5%)
  - 완두콩 (반응률: 5%)
  - 렌즈콩 (반응률: 5%)
  - 대두 (반응률: 5%)

- 성분: 잣  
대체 식품: 정보 없음  
교차반응군: 정보 없음

- 성분: 복숭아  
대체 식품: 타 과일  
교차반응군: 정보 없음

- 성분: 토마토  
대체 식품: 감자, 쌀, 우유  
교차반응군:
  - 그 외 곡류 (반응률: 20%)
  - 호밀 (반응률: 20%)
  - 보리 (반응률: 20%)

--- [개인 맞춤 알림] ---

주의! 'test'님께서 등록하신 '우유' 성분이 포함되어 있습니다!

- 회원 탈퇴:

'(8) 회원 탈퇴' 메뉴를 선택하면, 시스템은 정말로 삭제할 것인지 재차 확인한다.

'y'를 입력하면, 보안을 위해 현재 계정의 비밀번호를 다시 한번 입력하도록 요구한다.

비밀번호가 일치하면 성공적으로 회원 정보가 삭제되었다는 메시지가 출력되고, 시스템은 자동으로 로그아웃 상태가 된다.

```
-----
선택하세요 ==> 8

=== (8) 회원 탈퇴 ===
'test' 계정을 정말 삭제하시겠습니까? 되돌릴 수 없습니다. (y/n): y
비밀번호를 입력하여 본인인증을 완료하세요: testtest

회원 탈퇴가 성공적으로 처리되었습니다. 이용해주셔서 감사합니다.
```

- 관리자 메뉴 접근:

관리자 계정으로 로그인하면, 일반 사용자와 다릴 사용자 메뉴에 '(6) 관리자 메뉴'가 나타난다.

해당 메뉴를 선택하면, '전체 사용자 목록 보기', '최다 등록 알레르기 TOP 5' 등 관리자 전용 기능 목록이 있는 서브 메뉴로 진입한다.

```
선택하세요 ==> 7

=== (7) 로그인 ===
아이디: admin
비밀번호: admin1234

'admin'님, 환영합니다!

===== 식품 알레르기 확인 프로그램 =====
(admin 님, 환영합니다!)

--- 식품 정보 ---
(1) 식품 알레르기 조회
(2) 제품 정보 추가
(3) 식품 알레르기 정보 추가
(4) 식품 알레르기 정보 수정
(5) 제품 정보 삭제

--- 사용자 메뉴 ---
(6) 관리자 메뉴
(7) 로그아웃

(0) 프로그램 종료

-----
선택하세요 ==> █
```

- 데이터 분석 기능 실행:

관리자 메뉴에서 '(2) 최다 등록 알레르기 TOP 5'를 선택한다. 시스템은 User\_Allergies와 Allergy 테이블의 데이터를 JOIN하고 GROUP BY와 COUNT를 통해 집계하며, 사용자들이 가장 많이 등록한 알레르기 순위를 동적으로 계산하고 출력한다.

```
=== 관리자 메뉴 ===
(1) 전체 사용자 목록 보기
(2) 최다 등록 알레르기 TOP 5
(3) 메인 메뉴로 돌아가기
선택: 2

--- 최다 등록 알레르기 TOP 5 ---
1위: 우유 (3명)
2위: 땅콩 (2명)
3위: 난류 (1명)
4위: 새우 (1명)
5위: 대두 (1명)
```

## 5. 결론

### 5.1. 프로젝트 요약 및 성과

본 프로젝트는 특정 식품에 알레르기 정보를 확인하기 위해 여러 채널을 직접 방문해야 하는 소비자의 불편함을 해결하고자 시작되었다. 이를 위해, 웹 크롤링을 통해 분산된 데이터를 수집하고, 정규화된 관계형 데이터베이스를 구축하였으며, 사용자에게 맞춤형 정보를 제공하는 Python 기반 콘솔 애플리케이션을 성공적으로 개발하였다. 프로젝트를 통해 달성한 핵심적인 성과는 다음과 같다.

**첫째, 체계적인 관계형 데이터베이스를 구축하였다.**

사용자 정보, 식품 정보, 그리고 이들 간의 복잡한 다대다 관계를 해소하기 위해 총 10개의 테이블로 구성된 스키마를 설계하였다. 외래키와 ON DELETE CASCADE 제약조건을 적극적으로 활용하여 데이터의 무결성과 일관성을 확보하였으며, 이는 향후 기능 확장에 유연하게 대처할 수 있는 안정적인 기반이 된다.

**둘째, 데이터 수집부터 분석까지 전체 파이프라인을 구현하였다.**

단순히 주어진 데이터를 활용하는 것을 넘어, AJAX 통신과 OCR 등 기술적 난관을 해결하며 데이터를 직접 수집하였다. 이렇게 수집된 데이터를 정제하여 DB에 적재하고, 사용자는 이를 검색 및 개인화된 알림으로 활용하며, 관리자는 GROUP BY를 이용한 통계 분석으로 새로운 인사이트를 도출할 수 있다. 이는 데이터 처리의 전체 사이클을 직접 구현했다는 점에서 큰 의미를 가진다.



셋째, 보안과 개인화를 고려한 사용자 중심 기능을 개발하였다.

bcrypt를 이용한 비밀번호 해싱, 역할 기반 접근 제어, 비밀번호 재확인을 통한 안전한 회원 탈퇴 등 시스템의 보안성을 높였다. 또한, 사용자가 자신의 알레르기 정보를 등록하고 이를 바탕으로 개인화된 경고를 받는 기능은 단순 정보 제공을 넘어 사용자의 안전이라는 프로젝트 본연의 목표를 달성하는 핵심적인 성과이다.

## 5.2. 고찰 및 향후 개선 방안

본 프로젝트를 통해 데이터베이스 설계, 웹 크롤링, OCR 등 다양한 기술을 성공적으로 적용하고 문제 해결 능력을 기를 수 있었다. 하지만 제한된 시간과 자원으로 인해 다음과 같은 아쉬움이 남으며, 향후 다음과 같은 방향으로 시스템을 개선할 수 있을 것이다.

- 데이터 소스 확대 및 자동화:

현재는 '오투기몰'이라는 단일 소스에서 데이터를 수집하였으나, 더 많은 사용자를 위해 'CJ', '풀무원' 등 타 식품 제조사의 데이터를 추가로 수집하여 데이터베이스의 정보 범위를 확장할 필요가 있다. 또한, 주기적으로 신제품 정보를 자동으로 크롤링하여 DB에 업데이트하는 스케줄링 기능을 도입하여 데이터의 최신성을 유지할 수 있다.

- UI 개선:

현재의 콘솔 기반 인터페이스는 핵심 기능을 시연하고 검증하는 데는 효과적이었으나, 일반 사용자의 접근성이 떨어진다. 향후 웹 프레임워크나 모바일 애플리케이션으로 개발하여, 모든 사용자가 쉽게 접근하고 시각적으로 정보를 확인할 수 있는 GUI를 제공할 필요가 있다.

- OCR 엔진 고도화:

Tesseract OCR의 인식률 한계를 극복하기 위해 시도했던 PaddleOCR의 도입을, CUDA 버전 등 개발 환경 문제를 해결한 뒤 재시도하여 데이터의 정확성을 한 단계 더 향상시킬 수 있다.

- 사용자 편의 기능 추가:

사용자의 검색 로그를 저장하여 '최근 검색한 제품' 목록을 보여주거나, 다른 사용자들이 많이 검색하는 제품을 추천하는 등 사용자 경험을 증진시킬 수 있는 다양한 부가 기능을 추가할 수 있다.