

오픈소스 기초설계(나) 5팀 ^{off}

채식주의를 실천하는 사람들을 위한 맞춤 제품 정보 제공 웹사이트



문지환(팀 리더_20213069)



강민혁(20211720)



김연수(20211728)



반세현(20213073)

INDEX

☞ 주제 선정 계기
및 목표

☞ 해결 방안

☞ 필요 기술

☞ 웹 구조

☞ 결과 및 향후 개
발 방향

☞ Reference

☞ Q&A

Index

01

주제 선정 계기 및 목표

02

해결 방안

03

필요 기술

04

웹 구조

05

결과 및 향후 개발 방향

INDEX

1 주제 선정 및 목표

해결 방안

필요 기술

웹 구조

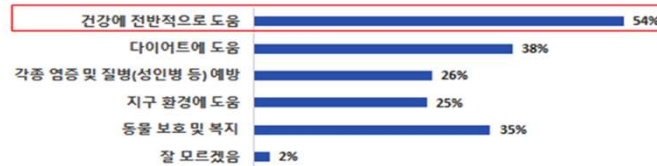
결과 및 향후 개발
방향

Reference

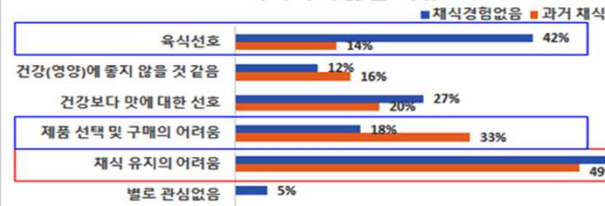
Q&A

1 주제 선정 계기 및 목표

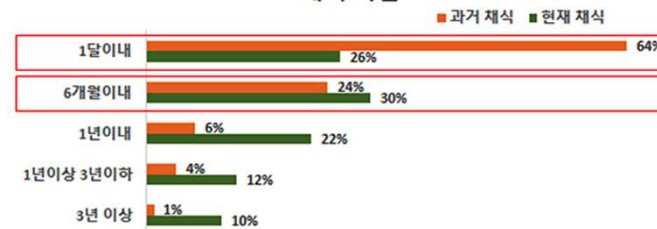
채식의 장점



채식하지 않는 이유



채식 기간



목표: 이미 채식주의를 실천하고 있는 사람들에게 쉬운 채식 생활을 장려하고, 채식주의를 시작하려고 하는 사람들에게 진입 장벽을 낮춰줄 웹사이트를 기획

INDEX

▣ 주제 선정 및 목표

▣ 해결 방안

▣ 필요 기술

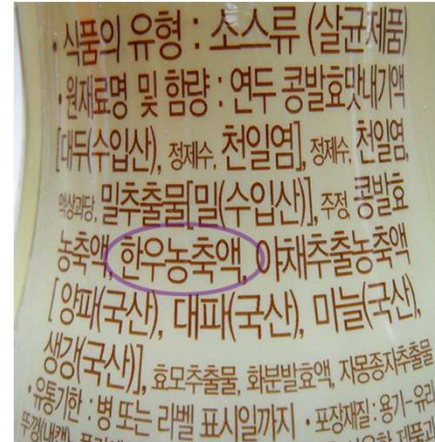
▣ 웹 구조

▣ 결과 및 향후 개발
방향

▣ Reference

▣ Q&A

2. 해결 방안



문제점

1. 비건 상품을 이용할 때 원재료명을 하나하나 찾아야 되는 점
2. 비건 상품들이 한 곳에 모여있지 않은 점

해결 방안

1. 시중의 상품들 중 비건에게 적합한 상품들을 모아서 정리, 검색 환경 구축
2. 더 나아가 채식주의의 단계와 원재료명에 따라 상품들을 분류

Veganary

Vegan + Dictionary

INDEX

≡ 주제 선정 및 목표

≡ 해결 방안

≡ 필요 기술

≡ 웹 구조

≡ 결과 및 향후 개발
방향

≡ Reference

≡ Q&A

3 필요 기술(설계 모델)



Frontend: HTML5, CSS3로 웹을 디자인하고 JAVASCRIPT로 여러 기능 구현
Backend: MySQL로 데이터베이스를, Node.js로 서버를 구축

JAVASCRIPT를 활용하여 모바일 상에서도 사용 가능한 반응형 웹으로 설계
제품명과 상점 이름 뿐만 아니라 원재료명 검색이 가능한 기능이 작동하도록 설계

INDEX

▣ 주제 선정 및 목표

▣ 해결 방안

▣ 필요 기술

▣ 웹 구조

▣ 결과 및 향후 개발
방향

▣ Reference

▣ Q&A

3 필요 기술(실제 구현)

BeautifulSoup



Selenium

python™

식품 정보 데이터: 파이썬 기반의 웹 스크래핑을 위한 라이브러리인 BeautifulSoup과 Selenium로 올리브영 홈페이지에서 스크래핑하고 XLSX파일로 저장



OpenRefine

CSV나 XLSX파일로 저장된 데이터를 쉽게 필터링할 수 있는 오픈소스 프로젝트인 OpenRefine로 스크래핑한 데이터를 원재료명에 따라 분류하여 채식주의에 적합한 제품과 아닌 제품을 구분



JAVASCRIPT를 활용하여 구현한 기능

1. 제품의 이름, 가격, 원재료명 등이 들어있는 데이터를 바로 웹 화면으로 진열
2. 메인 페이지에서 마우스 스크롤을 한번만 내려 매끄럽게 화면 첫 페이지 전체가 내려가서 검색창을 띄우는 기능
3. 제품명으로 검색하여 제품 데이터를 불러오는 기능

INDEX

≡ 주제 선정 및 목표

≡ 해결 방안

≡ 필요 기술

≡ 웹 구조

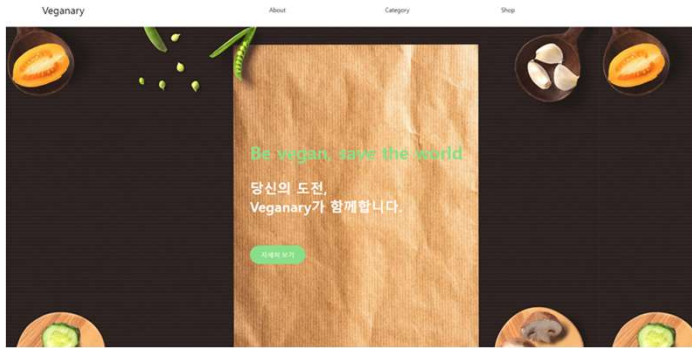
≡ 결과 및 향후 개발
방향

≡ Reference

≡ Q&A

4 웹 구조

메인 페이지



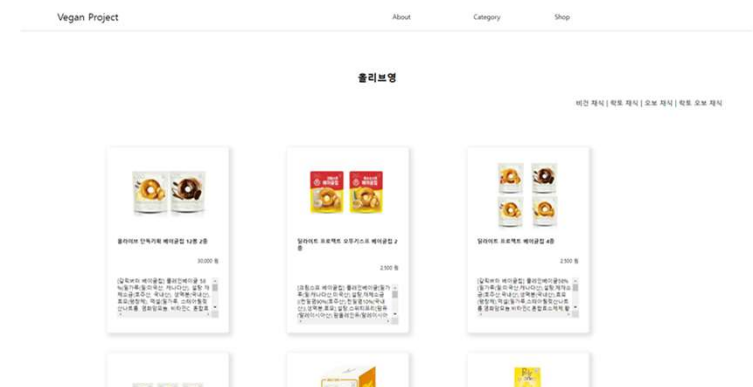
검색 페이지



소개 페이지



식품 진열 페이지



INDEX

▣ 주제 선정 및 목표

▣ 해결 방안

▣ 필요 기술

▣ 웹 구조

▣ 결과 및 향후 개발 방향

▣ Reference

▣ Q&A

5 결과 및 향후 개발 방향

비건 관련 제품들 판매가 중점인 기존의 서비스들

VEGANING

장바구니

VEGEFOOD
EST.1998

홈 SHOP 먹거리 라이프 지구 지식 커뮤니티

비건 가이드

테마쇼핑

콩고기 · 밀고기 쇼핑

그외 비건상품 쇼핑

알뜰코너

이벤트

결과

1. 다른 비건 관련 서비스들과는 차별된 시중의 제품들에 대한 정보를 제공
2. 채식주의를 실천하는 과정에서 생기는 문제점을 해결해주는 웹 사이트를 설계

향후 개발 방향

1. 다른 시중의 브랜드나 가맹점의 데이터를 받아 데이터베이스와 백엔드 서버를 구축하여 클라이언트와 active한 interaction을 할 수 있는 하나의 비건 인포(info) 플랫폼으로 발전
2. 상품명, 상점 이름, 원재료명 등으로 다양하게 검색 가능한 기능으로 개선
3. 상단 메뉴 바의 Shop에 비건 제품 판매 페이지를 추가하거나 비건 식당 추천 페이지를 새로 만든다면 더 많은 수요가 있을 것으로 예상

INDEX

☰ 주제 선정 및 목표

☰ 해결 방안

☰ 필요 기술

☰ 웹 구조

☰ 결과 및 향후 개발
방향

☰ Reference

☰ Q&A

REFERENCE

1. 식습관 통계자료: 채식 식습관 및 채식주의 관련 인식 조사(2018), Embrain
2. 크롤링 자료: 파이썬 코딩 무료 강의(활용편3)- 웹 크롤링? 웹 스크래핑!, 나도코딩, <https://youtu.be/yQ20jZwDjTE>
3. 크롤링 자료: 파이썬으로 웹 크롤러 만들기(2016), 라이언 미첼, 한빛미디어
4. 프론트 자료: w3schools.com
5. 다른 vegan 서비스: <https://veganing.co.kr/>, <http://www.vegefood.co.kr/>
6. 박현범, 'Lazy Loading'과 반응형 웹을 적용한 컴포넌트 기반의 웹 어플리케이션 개발 툴 구현, 한밭대학교 정보통신전문대학원, 2019

APPENDIX(0)

팀원별 역할 분담

문지환(팀 리더): 프론트엔드 개발, ppt 제작, 보고서 작성, 발표

강민혁: 데이터 크롤링, ppt 제작, 보고서 작성

김연수: 아이디어 제공, 웹 배포, ppt 제작, 보고서 작성

반세현: 프론트엔드 개발, ppt 제작, 보고서 작성

웹 URL

Veganary 웹 URL: <https://veganary.netlify.app/>

팀 깃허브 URL: https://github.com/Supermoon-JH/OpenSource_Project

APPENDIX(1)

BeautifulSoup을 활용한 데이터 스크래핑

```
1 import requests
2 from bs4 import BeautifulSoup
3
4 url1 = "https://www.oliveyoung.co.kr/store/display/getMCategoryList.do?dispCatNo=100000200020023&isLoginCnt=0&aShowCnt=0&bShowCnt=0&cShowCnt=0&gateCd=Drawer&trackingCd=Cat1000002"
5 res = requests.get(url1)
6 res.raise_for_status()
7
8 soup = BeautifulSoup(res.text, "lxml")
9
10 names = soup.find_all("p", attrs={"class": "tx_name"})
11 price_infos = soup.find_all("span", attrs={"class": "tx_cur"})
12 image_infos = soup.find_all("div", attrs={"class": "prd_info"})
13
14 for name, price, info in zip(names, price_infos, image_infos):
15     print(name.get_text())
16     price = price.span.get_text()
17     print(price+"원 ")
18     image = info.img["src"]
19     print(image)
20
```

APPENDIX(2)

Selenium을 활용한 데이터 스크래핑

```
1 import time
2
3 from openpyxl import Workbook
4 from selenium import webdriver as wd
5 from selenium.webdriver.chrome.service import Service
6 from selenium.webdriver.common.by import By
7
8
9 def main():
10     service = Service(
11         r"C:\Users\dkn1\Desktop\vegan2\vegan\py\chromedriver.exe")
12     options = wd.ChromeOptions() # 크롬 브라우저 실행 시 발생하는 경고 메시지
13     options.add_experimental_option("excludeSwitches", ["enable-logging"])
14     # options.add_argument("--disable-javascript")
15     # options.add_argument('headless') # 브라우저를 띄우지 않음
16     driver = wd.Chrome(service=service, options=options)
17     data = []
18
19     write_wb = Workbook() # 새 워크북 생성
20     write_ws = write_wb.active # 현재 활성화된 sheet 가져옴
21
22     write_ws.append( # 첫 번째 행에 추가
23         ["prd_brand", "prd_name", "prd_price", "prd_img", "prd_detail"])
24     # for prd_brand, prd_name, prd_price, prd_img, prd_detail in data:
25     #     write_ws.append([prd_brand, prd_name, prd_price, prd_img, prd_detail])
26
```

```
27 for i in range(1, 28): # 페이지 수가 총 27페이지
28     driver.get(
29         'https://www.oliveyoung.co.kr/store/display/getMCategoryList.do?dispCatNo=10000200020023&fltDispCatNo=8&prdSort=01&pageIc
30         i)
31     for j in range(2, 8): # ul 태그가 2~7까지
32         for k in range(1, 5): # li 태그가 1~4까지
33             driver.find_element(By.XPATH,
34                                 '/html/body/div[3]/div[7]/div[1]/ul[{}]/li[{}]/div/a'.format(j, k)).click()
35             driver.implicitly_wait(5) # 로딩이 다 되면 5초가 지나지 않아도 대기 종료
36             prd_brand = driver.find_element(
37                 By.XPATH, '/html/body/div[3]/div[7]/div/div[2]/div/p[1]').text
38             prd_name = driver.find_element(
39                 By.XPATH, '/html/body/div[3]/div[7]/div/div[2]/div/p[2]').text
40             prd_price = driver.find_element(By.XPATH,
41                                             '/html/body/div[3]/div[7]/div/div[2]/div/div[1]/span/strong').text
42             prd_img = driver.find_element(By.XPATH,
43                                           '/html/body/div[3]/div[7]/div/div[2]/div[1]/div[1]/img').get_attribute(
44                 'src')
45
46             time.sleep(10) # 로딩이 늦을 시를 대비해 무조건 10초 대기
47
```

```
48 btn = driver.find_element(
49     By.XPATH, '/html/body/div[3]/div[7]/div/ul/li[2]/a') # href 태그에 href 주소가 없어서 click 대신에 btn 도입
50 driver.execute_script("arguments[0].click();", btn)
51 # enter 기능을 javascript로 구현
52
53 prd_detail = driver.find_element(
54     By.XPATH, '/html/body/div[3]/div[7]/div/div[7]/div[2]/div[6]/dd').text
55 item = [prd_brand, prd_name, prd_price, prd_img, prd_detail]
56 data.append(item)
57 print(item) # 데이터베이스에서 확인
58 write_ws.append(item) # sheet에 데이터 추가
59 driver.back() # 웹에서 들어간 페이지에서 다시 나올
60 write_wb.save('./data.xlsx') # sheet를 xlsx 파일로 저장
61
62
63 if __name__ == "__main__":
64     main()
65
```

APPENDIX(3)

OpenRefine을 활용한 데이터 정제

OpenRefine data backup.xlsx Permalink

Facet / Filter Undo / Redo 0 / 0 45 matching records (122 total) Extensions: Wikidata

Refresh Reset All Remove All

☐ case sensitive ☐ regular expression

pr_d_detail invert reset

젤라틴 ☐ case sensitive ☐ regular expression

pr_d_detail invert reset

황태 ☐ case sensitive ☐ regular expression

pr_d_detail invert reset

페이지 ☐ case sensitive ☐ regular expression

pr_d_detail invert reset

상세 ☐ case sensitive ☐ regular expression

pr_d_detail invert reset

계란 ☐ case sensitive ☐ regular expression

pr_d_detail invert reset

우유 ☐ case sensitive ☐ regular expression

pr_d_detail invert reset

밀크 ☐ case sensitive ☐ regular expression

pr_d_detail invert reset

분유 ☐ case sensitive ☐ regular expression

pr_d_detail invert reset

가다랑어 ☐ case sensitive ☐ regular expression

pr_d_brand	pr_d_name	pr_d_price	pr_d_img	pr_d_detail
5. 달라이트 프로텍트	달라이트 프로텍트 존독한 망고알갱이 5입기릭	14,500	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0014/A00000014582206ko.jpg?l=ko	망고(베트남산) 95%, 설탕 4%, 구연산, 푸수아칼산(산화방지제) *아칼산류 함유(알라르기 유발물질)
6. 포지탁노	포지탁노 레몬캔디 200g	7,900	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0012/A00000012769502ko.jpg?l=ko	설탕, 포도당시럽, 포도당, 구연산, 천연향료, 심황색소
8. 달라이트 프로텍트	달라이트 프로텍트 근약위타일리 3종	1,900	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0015/A00000015419401ko.jpg?l=ko	<자두맛> 정제수, 에리스리톨, 혼합제제(카라기난, 구연산, 산나트륨, 근약분말(중국산), 잔탄검), 합성향료(자두향), 구연산, 스트로베리향료(영양배우색소, 크지발주출출소, 프로폴리스향료, 정제수), 구연산, 산나트륨, 꽃산향료, 자두농축액(미국산/ 자두 100%), 비타민 C, 염화마그네슘(해당성분수 추출물 100%), 감미료(수크랄로스, 아세설팜알루미늄), 피쉬콜라겐
10. 밀키오	밀키오 그릭요거트 (3가지 맛 플레인/알기/포도) (130g*6개입)	33,000	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0015/B00000015907002ko.jpg?l=ko	<정제수> 정제수, 에리스리톨, 혼합제제(카라기난, 구연산, 산나트륨, 근약분말(중국산), 잔탄검), 합성향료(자두향), 구연산, 스트로베리향료(영양배우색소, 크지발주출출소, 프로폴리스향료, 정제수), 구연산, 산나트륨, 꽃산향료, 자두농축액(미국산/ 자두 100%), 비타민 C, 염화마그네슘(해당성분수 추출물 100%), 감미료(수크랄로스, 아세설팜알루미늄), 피쉬콜라겐
12. 달라이트 프로텍트	달라이트 프로텍트 바삭한 고구마칩 50g	2,500	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0010/A00000010338205ko.jpg?l=ko	<사과맛> 정제수, 에리스리톨, 혼합제제(카라기난, 구연산, 산나트륨, 근약분말(중국산), 잔탄검), 사과농축액(중국산/사과 100%), 구연산, 엑스트린, 염화마그네슘(해당성분수 추출물 100%), 꽃산향료, 비타민 C, 합성향료(사과향), 감미료(수크랄로스, 아세설팜알루미늄), 피쉬콜라겐
13. 멜로슬로	멜로슬로 카키오칩 25g	3,000	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0012/A00000012647505ko.jpg?l=ko	원유(유기농, 국산), 원유(무함상제, 국산), 설탕(유기농, 태국산), 유산균(연마르산), 당류가공품(알기, 포도, 국산), 합성향료(알기향, 포도향)
15. 퍼지탁	퍼지탁 자일리톨 캔디 플레인타일 40g	3,500	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0001/A00000001634002ko.jpg?l=ko	상고구마(국내산) 84.49%, 설탕, 말줄레인유(말레이시아산)
20. 리콜라	리콜라슈거프리레온40g	2,000	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0012/A00000012651602ko.jpg?l=ko	카카오페이트 35%, 코코넛 과육 35%, 코코넛꽃꿀 30%
21. 잇플	잇플 그린 24C	4,500	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0013/A00000013216503ko.jpg?l=ko	■ 자일리톨 99.7%, 민트향(천연향료) 0.3%
27. 기라일리	기라일리 초콜릿 117g 4종-글라디기	12,000	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0014/A00000014734602ko.jpg?l=ko	이소말트, 정제수, 구연산, 레몬향, 리콜라슈거추출물 0.16% (그린 0.1%), 천연향료(레몬향) 0.21%, 퍼지탁향 0.04%, 연꽃향, 감미료(사과향-과일향, 말줄레인유, 아세설팜알루미늄, 잔탄검(베타카르틴))
28. 룩트	[업제배출] 룩트 요거트 100g * 6개입 3종 중 택1 (미크/마일드)	21,760	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0015/B00000015320901ko.jpg?l=ko	*아보카도오일(멕시코산), 천연향료(박하향), 대두레시틴, 파슬리오일, 로즈마리오일(독일산)
32. 퍼지탁	퍼지탁 자일리톨 캔디 복숭아향 40g	3,900	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0012/A00000012141002ko.jpg?l=ko	*합성향료: 변성전분, 글리세린, 카라기난
33. 달라이트 프로텍트	달라이트 프로텍트 존독한 망고알갱이 80g	3,800	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0014/A00000014453608ko.jpg?l=ko	[대두 함유]
34. 레이즈	레이즈 클래식 감자칩 42.5g	900	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0015/A00000015407901ko.jpg?l=ko	연스위트초콜릿, 설탕, 코코아버터, 바닐라추출물, 유화제
35. 씨제이제일제당	CJ 맛밤 80g	2,000	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0001/A00000001161102ko.jpg?l=ko	아이슬란드 요거트 따크 100g : 원유(세균수 기준 1급A, 국산) 99.96%, 유산균
36. 잇플	잇플 핑크 24	4,500	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0014/A00000014187001ko.jpg?l=ko	아이슬란드 요거트 마일드 100g : 원유(세균수 기준 1급A, 국산) 99.96%, 유산균
38. 달라이트 프로텍트	달라이트 프로텍트 바삭한 자색고구마칩 50g	2,500	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0012/A00000012794603ko.jpg?l=ko	자일리톨 99.1% / DL-사과산 0.6% / 복숭아향(천연향료) 0.3%
40. 달라이트 프로텍트	달라이트 프로텍트 바삭한 완콩칩 3종 중 택1(고구마칩/자색고구마칩/느타리바섯칩)	2,500	https://image.oliveyoung.co.kr/uploads/images/goods/550/10/0000/0015/A00000015633201ko.jpg?l=ko	망고(베트남산) 95%, 설탕 4%, 구연산, 푸수아칼산(산화방지제)

감자82.04%, 식물성유지(해바라기유, 옥수수유, 카놀라유), 정제수

반향100%

*상산 및 유통 시점에 따라 원산지 변경 전 제품이 혼용되어 발송 가능한 점 양해 부탁드립니다.

*올리브유(스페인산), 천연향료(박하향), 합성향료(복숭아향1, 복숭아향2), 대두레시틴, 로즈마리오일(독일산), 천연향료(유칼립투스향)

*합성향료: 변성전분, 글리세린, 카라기난, 이산화탄소(착색소), 지장적색소

[대두 함유]

상자색고구마 82%(국내산), 설탕, 말줄레인유(말레이시아산)

[바삭한 고구마칩]

상고구마(국내산) 84.49%, 설탕, 말줄레인유(말레이시아산)

[바삭한 자색고구마칩]

상자색고구마 82%(국내산), 설탕, 말줄레인유(말레이시아산)

[바삭한 느타리바섯칩]

APPENDIX(4)

JAVASCRIPT를 활용한 데이터 진열

```
1 window.onload = function () {  
2  
3     console.log(2)  
4     let xhr = new XMLHttpRequest();  
5     xhr.open("GET", "../javascript/vegan.xlsx", true);  
6     xhr.responseType = "blob";  
7     xhr.onload = function (e) {  
8         let file = this.response;  
9         let reader = new FileReader();  
10        if (reader.readAsBinaryString(file)) {  
11            reader.onload = function (e) {  
12  
13                ProcessExcel(e.target.result);  
14            };  
15            reader.readAsBinaryString(file);  
16        }  
17    };  
18    xhr.send();  
19 }  
20  
21 function ProcessExcel(data) {  
22     let workbook = XLSX.read(data, {type: 'binary'});  
23     let firstSheet = workbook.SheetNames[0];  
24     let excelRows = XLSX.utils.sheet_to_row_object_array(workbook.Sheets[firstSheet]);  
25  
26     let items = document.getElementById('items')  
27  
28     for (let i = 0; i < excelRows.length; i++) {  
29         console.log(1)  
30         let item = document.createElement('div')  
31         item.setAttribute('class', 'item')  
32  
33         let prd_name = document.createElement('p')  
34         prd_name.innerHTML = excelRows[i].prd_name  
35         prd_name.style.fontWeight = 'bold'  
36  
37         let prd_price = document.createElement('p')  
38         prd_price.innerHTML = excelRows[i].prd_price + ' 원'  
39         prd_price.style.textAlign = 'right'  
40         prd_price.style.fontSize = '12px'  
41  
42         let prd_detail = document.createElement('p')  
43         prd_detail.innerHTML = excelRows[i].prd_detail  
44         prd_detail.style.maxHeight = '100px'  
45         prd_detail.style.overflow = 'scroll'  
46  
47         let prd_img = document.createElement('img')  
48         prd_img.setAttribute('src', excelRows[i].prd_img)  
49         item.appendChild(prd_img)  
50         item.appendChild(prd_name)  
51         item.appendChild(prd_price)  
52         item.appendChild(prd_detail)  
53         items.appendChild(item)  
54     }  
55 }
```


APPENDIX(5)

JAVASCRIPT를 활용한 검색 기능 구현

```
1 // 전역 변수 설정
2 let data;
3 let keyword;
4
5 window.onload = function () {
6     //search 라는 id 가진 태그에 change 이벤트 부여
7     document.getElementById('search').addEventListener('change', (e) => {
8         keyword = e.target.value // input 에다가 입력하면 전역변수 keyword 값이 바뀌도록 하는 부분
9     });
10    // search_btn 라는 id 가진 태그에 click 이벤트 부여
11    document.getElementById('search_btn').addEventListener('click', () => {
12        Search(); // 클릭하면 옆에 함수 호출
13    });
14
15    let xhr = new XMLHttpRequest();
16    xhr.open("GET", "vegan.xlsx", true);
17    xhr.responseType = "blob";
18    xhr.onload = function (e) {
19        let file = this.response;
20        let reader = new FileReader();
21        if (reader.readAsBinaryString) {
22            reader.onload = function (e) {
23                let workbook = XLSX.read(e.target.result, {type: 'binary'});
24                let firstSheet = workbook.SheetNames[0];
25                let excelRows = XLSX.utils.sheet_to_row_object_array(workbook.Sheets[firstSheet]);
26                data = excelRows
27            };
28            reader.readAsBinaryString(file);
29        }
30    };
31    xhr.send();
32 }
```

```
34 // 검색하는 함수
35 function Search() {
36     // 일단 변수 선언 (같은 할당 x)
37     let items;
38     // 이미 검색을 한번 해서 검색 결과가 화면에 있으면 지워주고 없으면 새로 만들어서 진행.
39     if (!document.getElementById('items')) {
40         items = document.createElement('div')
41         items.setAttribute('id', 'items')
42     } else {
43         document.getElementById('items').remove()
44         items = document.createElement('div')
45         items.setAttribute('id', 'items')
46     }
47
48     for (let i = 0; i < data.length; i++) {
49         if (data[i].prd_name.split(' ').includes(keyword)) {
50             let item = document.createElement('div')
51             item.setAttribute('class', 'item')
52
53             let prd_name = document.createElement('p')
54             prd_name.innerHTML = data[i].prd_name
55             prd_name.style.fontWeight = 'bold'
56
57             let prd_price = document.createElement('p')
58             prd_price.innerHTML = data[i].prd_price + ' 원'
59             prd_price.style.textAlign = 'right'
60             prd_price.style.fontSize = '12px'
61
62             let prd_detail = document.createElement('p')
63             prd_detail.innerHTML = data[i].prd_detail
64             prd_detail.style.maxHeight = '100px'
65             prd_detail.style.overflow = 'scroll'
66
67             let prd_img = document.createElement('img')
68             prd_img.setAttribute('src', data[i].prd_img)
69             item.appendChild(prd_img)
70             item.appendChild(prd_name)
71             item.appendChild(prd_price)
72             item.appendChild(prd_detail)
73             items.appendChild(item)
74         }
75     }
76
77     document.getElementById('result').appendChild(items)
78
79 }
80 }
```

Q&A