



Sugar high

200785 김은서



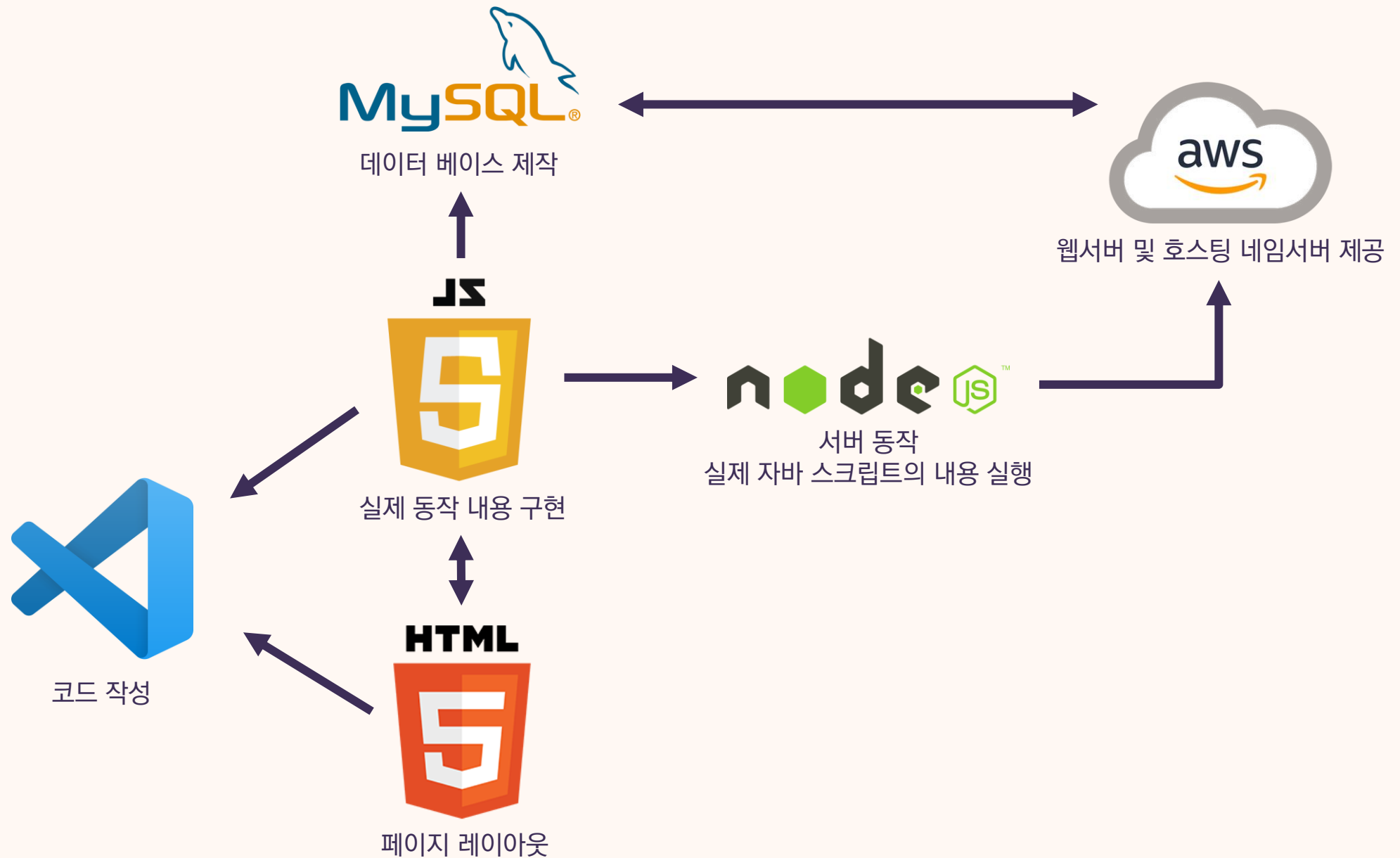
아이디어

- 최근 들어 마트에 흔히 발견되는 각종 제로 슈가 제품
 - 건강에 대한 사람들의 관심이 점차 증가
 - 제로 슈가 제품이 있는 이유: 혈당 관리 !
 - 영양성분표가 나와있는 공산품과 달리, 밖에서 사먹는 음식들은 안에 무엇이, 얼마나 들어있는지 잘 모르고 그냥 먹게 되는 문제
 - 특히, 카페는 음료와 디저트 위주라 당류 함량이 높을 것이라고 예상되나, 대부분의 카페 내부에서 관련 정보를 찾아볼 수 없음.
-
- 즉, 평소 자주 가는 카페와 음료 위주로 데이터베이스를 구성하여 각 메뉴를 검색 하면 당류와 칼로리 등의 정보를 제공하고자 한다.

응용시스템

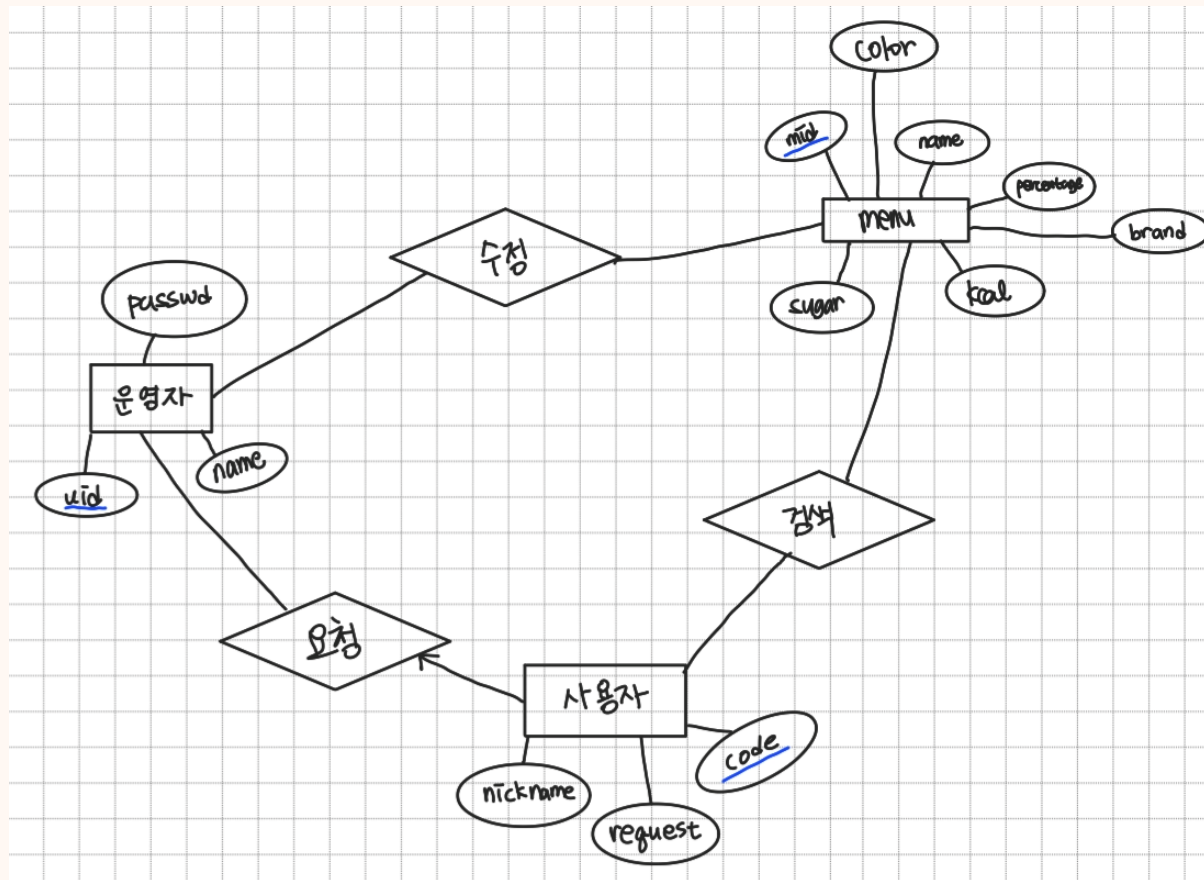
- mySQL 데이터베이스 사용
- VSCode(Visual Studio Code)로 HTML 및 javascript 코드 작성
- node.js 를 통해 서버 구현
- amazon aws를 통한 웹서버 구축 및 호스팅

시스템 아키텍처



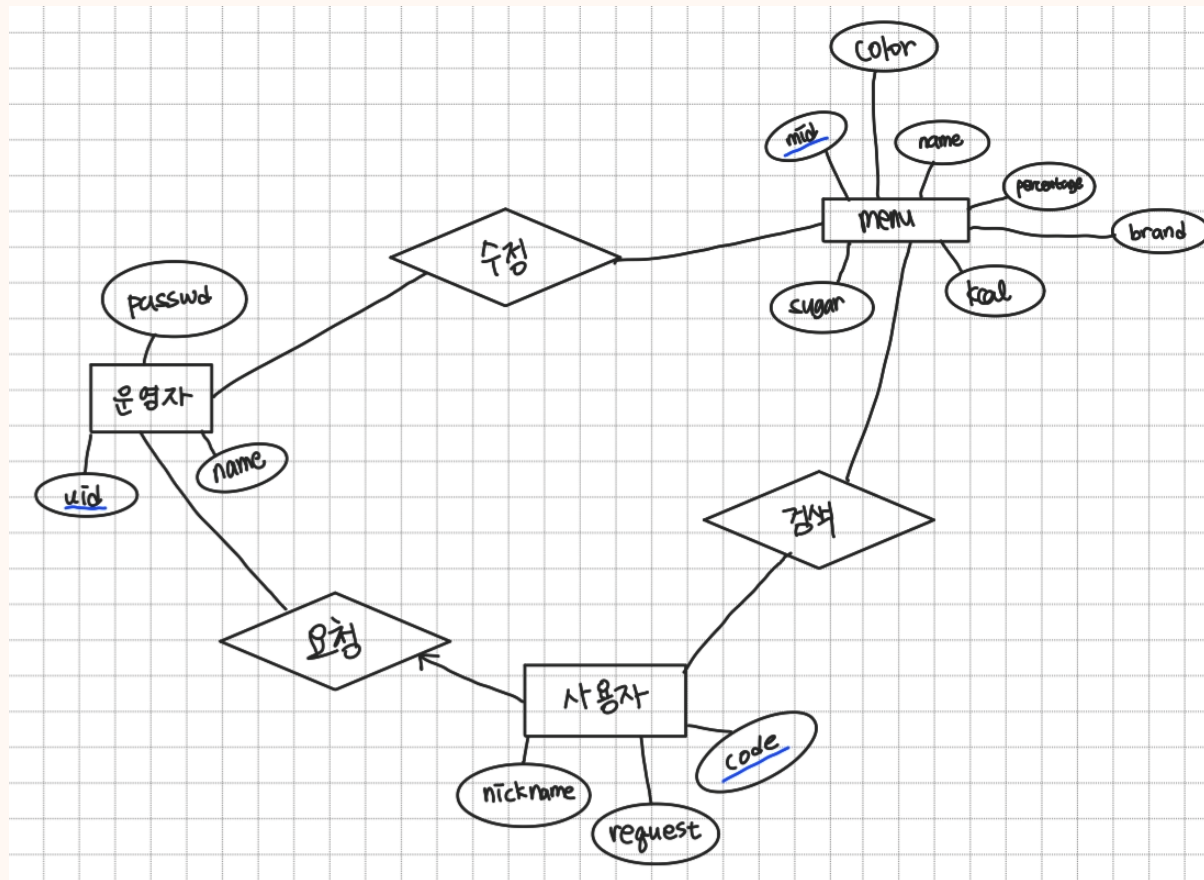
데이터베이스

ER-diagram



- menu : 음료의 개별 코드(mid), 브랜드(brand), 이름(name), 당류의 양(sugar), 칼로리(kcal), 당류 권장량 기준치(percentage), 색깔(color)
- 운영자(account) : 운영자 아이디(uid), 이름(name), 비밀번호(passwd)
- 사용자(community) : 인식 코드(code), 이름(nickname), 요청내용(request)

ER-diagram



- 운영자는 menu 데이터베이스를 수정할 수 있다.
- 사용자는 menu 데이터베이스를 검색할 수 있다.
- 사용자는 운영자에게 원하는 정보를 요청할 수 있다.
- 이때, 현재까지 운영자는 1명이므로 각 사용자는 운영자와 1:1 관계이다. (모든 사용자는 운영자 1명과 관계를 맺는다.)

schema – menu

1	mid	name	sugar	kcal	percentage	color	brand
2	1001	리저브나오	0	5	0	brown	스타벅스
3	1002	나이트로	10	80	20	brown	스타벅스
4	1003	나이트로	0	5	0	brown	스타벅스
5	1004	돌체 콜드	29	265	58	brown	스타벅스
6	1005	리저브 콜	0	5	0	black	스타벅스
7	1006	민트 콜드	23	100	46	brown	스타벅스
8	1007	바닐라 크	11	125	22	brown	스타벅스
9	1008	시그니처	0	25	0	black	스타벅스
10	1009	콜드브루	0	5	0	black	스타벅스
11	1010	콜드브루	41	505	82	ivory	스타벅스
12	1011	콜드브루	14	120	28	brown	스타벅스
13	1012	콜드브루	18	225	36	brown	스타벅스
14	1013	아이스커피	0	5	0	black	스타벅스
15	1014	슈크림라떼	41	345	82	brown	스타벅스
16	1015	아이스슈크	35	280	70	brown	스타벅스
17	1016	에스프레소	1	30	2	brown	스타벅스
18	1017	에스프레소	0	10	0	brown	스타벅스
19	1018	아이스 카	0	10	0	black	스타벅스
20	1019	카페 아메	0	10	0	black	스타벅스
21	1020	아이스 카	22	190	44	brown	스타벅스
22	1021	카라멜 마	22	200	44	brown	스타벅스
23	1022	아이스 카	9	115	18	brown	스타벅스

```
ALTER TABLE menu
MODIFY COLUMN mid INT PRIMARY KEY
```

399	5046	청포도 스	52.2	220.2	104.4	green	컴포즈
400	5047	패션후르츠	55.5	226	111	yellow	컴포즈
401	5048	타트체리	51.3	246.7	102.6	red	컴포즈
402	5049	자몽 에이	42.2	175.6	84.4	red	컴포즈
403	5050	레몬 에이	43.8	178.8	87.6	yellow	컴포즈

- mid(int): 각 메뉴를 구분할 코드, 브랜드 별 코드로 구분된 4자리 숫자, 중복비허용
- name(text): 메뉴 이름
- sugar(double): 당의 양
- kcal(double): 열량
- percentage(double): 하루 당류 권장량에서 해당 음료가 차지하는 당류의 비율
- color(text): 음료의 색깔
- brand(text): 음료를 판매하고 있는 브랜드

- 전남대학교 후문 기준 가까운 곳에 있는 6곳(스타벅스, 공차, 할리스커피, 매머드익스프레스, 더 벤티, 컴포즈커피)을 기준으로 데이터베이스 설계
- 각 브랜드별로 표기된 기준이 다르므로 메인 화면 및 검색 화면에 기준을 추가적으로 기입
- 당류의 비율은 국제기준 50g을 기준으로 표시
- sugar와 kcal은 모두 한 잔에 들어있는 양으로 통일, 잔의 사이즈를 바꿀 수 있는 경우에는 기본 사이즈(tall)에 맞추어 작성

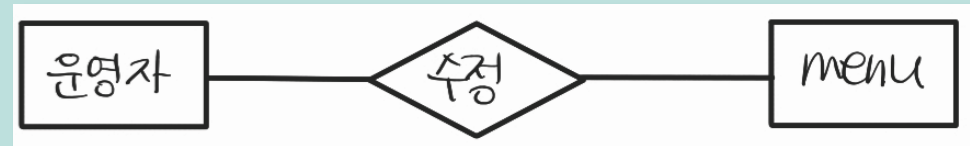
schema – account

비밀번호 입력 : 인증

	A	B	C
1	uid	name	passwd
2	1	root	

```
ALTER TABLE account  
MODIFY COLUMN uid INT PRIMARY KEY
```

- uid(int): 운영자를 구분할 아이디, 중복비허용
- name(text): 운영자 이름
- passwd(int): 숫자 비밀번호만 허용



- 운영자는 웹에서는 추가를 허용하지 않고, 아직 전체 sys에 대해 본인 혼자이므로, 사용자와 1대다, 내지는 1대1 관계를 형성
- 비밀번호 입력으로 수정페이지에 접속하여 기존 데이터베이스 수정 및 추가 가능
- 비밀번호 수정 역시 웹페이지로는 허용하지 않고, 직접 데이터베이스에 접속하여 수정

schema – community



	A	B	C	D	E	F
1	code	nickname	request			
2		1 akira	스타벅스 신메뉴도 추가해주세요			
3		2 sendoh	메가커피도 넣어주세요			
4		3 maki	공차 타로버블티(펄) 정보 재확인 부탁드립니다			

```
ALTER TABLE community  
MODIFY COLUMN code INT PRIMARY KEY
```

- **code(int)**: 사용자 식별코드, 중복비허용
- **nickname(text)**: 사용자 이름
- **request(text)**: 요청 내용

code :

이름 :

정보 요청 / 수정 등 요청 내용을 입력해주세요.

- 기본적으로 홈페이지는 로그인 없이 사용될 수 있고, 대부분의 사용자는 주로 요청사항 없이 검색 위주로 사용할 것으로 예상
- 여기서 말하는 사용자는 모든 사용자가 아닌 정보를 요청 및 수정을 질의하는 사용자만을 의미
- 요청사항이 관계의 속성이 아니라 사용자의 속성으로 들어간 이유는 웹페이지를 통해 운영자가 요청내용을 바로 확인하므로, 굳이 요청사항 스키마를 따로 구현할 필요가 없기 때문

코드 및 인터페이스

app.js

```
const express = require('express');
const app = express();
const port = 8080;
const qs = require('qs');
const bodyParser = require('body-parser');
const mysql = require('mysql');
const conn = { // mysql 접속 설정
  //
};

let connection = mysql.createConnection(conn); // DB 커넥션 생성
connection.connect(); // DB 접속
let sqls;
let param;
```

express: 웹 서버용 프레임워크 모듈
이외 qs, mysql 등 필요한 모듈 설치

```
//접속
v app.listen(port, () => {
  |   console.log('test')
  | })
```

```
//이미지 링크
✓ app.get('/logo', (req,res) => {
  |   res.sendFile(__dirname + "/logo_k_e.png");
  | });

✓ app.get('/srbutton', (req,res) => {
  |   res.sendFile(__dirname + "/srbutton.png");
  | });

✓ app.get('/combutton', (req,res) => {
  |   res.sendFile(__dirname + "/combutton.png");
  | });
```

```
//화면 링크(프론트)
✓ app.get('/', (req,res) => {
  |   res.sendFile(__dirname + "/sugarhigh.html");
  | });

✓ app.get('/search', (req,res) => {
  |   res.sendFile(__dirname + "/searchpage.html");
  | });

✓ app.get('/community', (req,res) => {
  |   res.sendFile(__dirname + "/community.html");
  | });

✓ app.get('/modify', (req,res) => {
  |   res.sendFile(__dirname + "/modify.html");
  | });

✓ app.get('/certification', (req,res) => {
  |   res.sendFile(__dirname + "/certificate.html");
  | });

✓ app.get('/writer', (req,res) => {
  |   res.sendFile(__dirname + "/writer.html");
  | });
```

로고 이미지, 검색 버튼,
게시판 버튼 이미지를 미리
만든 후 하이퍼링크 연결

슈가 하 이
SUGAR HIGH.

검색
게시판

- 사이트 내 영양정보 작성 기준

본 사이트는 스타벅스, 골라, 할리스커피, 덕엔티, 겔포즈커피, 데어드 익스프레스 메뉴에 대해서만 영양정보를 제공합니다.

신제품, 영양정보 변경에 대해서 주기적으로 업데이트를 하나 영양정보 수정 등 문의사항은 게시판을 이용해주시길 바랍니다.

하루 당류 총족 기준은 국제 기준인 50g을 기준으로 계산된 값입니다.

스타벅스: 에스프레소(22ml) / 더블샷(207ml) / 럽샷 코르타도(샷, 237ml) / 더 그린북 라인, 플립 섀넌(그린데, 473ml) / 이 외의 메뉴는 모두 풀(335ml)사이즈

공차: ice(473ml) / hot(414ml)

할리스 커피: 모든 메뉴 354ml

메머드 익스프레스: ice(20온즈, 약 590ml) / hot(16온즈, 약 473ml) / 메머드사이즈(32온즈, 약 946ml)

더벤티와 컴포즈 커피는 관련 표기 미확인 상태입니다

- 당류의 위험성

당류를 과잉 섭취할 경우 기억력의 중추인 해마를 위축시켜 혈관성 치매 위험을 높이고, 비만 위험도는 1.39배 높아지며, 천성 당뇨병을 넘어 다양한 암의 위험 또한 높인다.

신변에 향한 당분은 비만을 유발하고, 이에 따라 인슐린 저항성이 증가하면서 뇌기능을 저하시킨다. 기억력의 증추인 해마(측두엽의 안쪽에 위치하면서 기억을 저장하고, 공간 개념, 정서적인 행동을 조절하는 뇌조직)가 되고, 포도당의 혈중 수치가 높은 사람이 정상인 사람에 비해 해마 크기가 작아져 기억력이 떨어진다는 것이다. 또한, 과도한 설탕 섭취는 기분과 기억을 해치는 신경전달물질인 글루타민산과 신경염증반응을 일으키므로 우울증 위험을 증가시킨다. 잘 알려져 있듯이 우울증은 차례 별상 위험을 2~3배나 증가시킨다. 한 단이 들어 온것을 잘 자주 단을 잘 간성하고, 단을 잘 위험도 높여지게 되는 것이다.

출생과 사망률 간의 상관관계는 높은 편이지만, 특히 사망률은 매우 낮아진다. 고령과 고소득이 높은 편인 일본은 다른 선진국들처럼 일종의 노년층에 대한 보호정책을 시행하고 있다. 노년층의 사망률은 다른 선진국에 비해 낮은 편이다. 노년층의 사망률은 1990년 10.4%로, 2000년 10.1%로, 2005년 9.9%로, 2009년 9.7%로, 2010년 9.6%로, 2011년 9.5%로, 2012년 9.4%로, 2013년 9.3%로, 2014년 9.2%로, 2015년 9.1%로, 2016년 9.0%로, 2017년 8.9%로, 2018년 8.8%로, 2019년 8.7%로, 2020년 8.6%로, 2021년 8.5%로, 2022년 8.4%로, 2023년 8.3%로, 2024년 8.2%로, 2025년 8.1%로, 2026년 8.0%로, 2027년 7.9%로, 2028년 7.8%로, 2029년 7.7%로, 2030년 7.6%로, 2031년 7.5%로, 2032년 7.4%로, 2033년 7.3%로, 2034년 7.2%로, 2035년 7.1%로, 2036년 7.0%로, 2037년 6.9%로, 2038년 6.8%로, 2039년 6.7%로, 2040년 6.6%로, 2041년 6.5%로, 2042년 6.4%로, 2043년 6.3%로, 2044년 6.2%로, 2045년 6.1%로, 2046년 6.0%로, 2047년 5.9%로, 2048년 5.8%로, 2049년 5.7%로, 2050년 5.6%로, 2051년 5.5%로, 2052년 5.4%로, 2053년 5.3%로, 2054년 5.2%로, 2055년 5.1%로, 2056년 5.0%로, 2057년 4.9%로, 2058년 4.8%로, 2059년 4.7%로, 2060년 4.6%로, 2061년 4.5%로, 2062년 4.4%로, 2063년 4.3%로, 2064년 4.2%로, 2065년 4.1%로, 2066년 4.0%로, 2067년 3.9%로, 2068년 3.8%로, 2069년 3.7%로, 2070년 3.6%로, 2071년 3.5%로, 2072년 3.4%로, 2073년 3.3%로, 2074년 3.2%로, 2075년 3.1%로, 2076년 3.0%로, 2077년 2.9%로, 2078년 2.8%로, 2079년 2.7%로, 2080년 2.6%로, 2081년 2.5%로, 2082년 2.4%로, 2083년 2.3%로, 2084년 2.2%로, 2085년 2.1%로, 2086년 2.0%로, 2087년 1.9%로, 2088년 1.8%로, 2089년 1.7%로, 2090년 1.6%로, 2091년 1.5%로, 2092년 1.4%로, 2093년 1.3%로, 2094년 1.2%로, 2095년 1.1%로, 2096년 1.0%로, 2097년 0.9%로, 2098년 0.8%로, 2099년 0.7%로, 2100년 0.6%로, 2101년 0.5%로, 2102년 0.4%로, 2103년 0.3%로, 2104년 0.2%로, 2105년 0.1%로, 2106년 0.0%로, 2107년 0.0%로, 2108년 0.0%로, 2109년 0.0%로, 2110년 0.0%로, 2111년 0.0%로, 2112년 0.0%로, 2113년 0.0%로, 2114년 0.0%로, 2115년 0.0%로, 2116년 0.0%로, 2117년 0.0%로, 2118년 0.0%로, 2119년 0.0%로, 2120년 0.0%로, 2121년 0.0%로, 2122년 0.0%로, 2123년 0.0%로, 2124년 0.0%로, 2125년 0.0%로, 2126년 0.0%로, 2127년 0.0%로, 2128년 0.0%로, 2129년 0.0%로, 2130년 0.0%로, 2131년 0.0%로, 2132년 0.0%로, 2133년 0.0%로, 2134년 0.0%로, 2135년 0.0%로, 2136년 0.0%로, 2137년 0.0%로, 2138년 0.0%로, 2139년 0.0%로, 2140년 0.0%로, 2141년 0.0%로, 2142년 0.0%로, 2143년 0.0%로, 2144년 0.0%로, 2145년 0.0%로, 2146년 0.0%로, 2147년 0.0%로, 2148년 0.0%로, 2149년 0.0%로, 2150년 0.0%로, 2151년 0.0%로, 2152년 0.0%로, 2153년 0.0%로, 2154년 0.0%로, 2155년 0.0%로, 2156년 0.0%로, 2157년 0.0%로, 2158년 0.0%로, 2159년 0.0%로, 2160년 0.0%로, 2161년 0.0%로, 2162년 0.0%로, 2163년 0.0%로, 2164년 0.0%로, 2165년 0.0%로, 2166년 0.0%로, 2167년 0.0%로, 2168년 0.0%로, 2169년 0.0%로, 2170년 0.0%로, 2171년 0.0%로, 2172년 0.0%로, 2173년 0.0%로, 2174년 0.0%로, 2175년 0.0%로, 2176년 0.0%로, 2177년 0.0%로, 2178년 0.0%로, 2179년 0.0%로, 2180년 0.0%로, 2181년 0.0%로, 2182년 0.0%로, 2183년 0.0%로, 2184년 0.0%로, 2185년 0.0%로, 2186년 0.0%로, 2187년 0.0%로, 2188년 0.0%로, 2189년 0.0%로, 2190년 0.0%로, 2191년 0.0%로, 2192년 0.0%로, 2193년 0.0%로, 2194년 0.0%로, 2195년 0.0%로, 2196년 0.0%로, 2197년 0.0%로, 2198년 0.0%로, 2199년 0.0%로, 2200년 0.0%로, 2201년 0.0%로, 2202년 0.0%로, 2203년 0.0%로, 2204년 0.0%로, 2205년 0.0%로, 2206년 0.0%로, 2207년 0.0%로, 2208년 0.0%로, 2209년 0.0%로, 2210년 0.0%로, 2211년 0.0%로, 2212년 0.0%로, 2213년 0.0%로, 2214년 0.0%로, 2215년 0.0%로, 2216년 0.0%로, 2217년 0.0%로, 2218년 0.0%로, 2219년 0.0%로, 2220년 0.0%로, 2221년 0.0%로, 2222년 0.0%로, 2223년 0.0%로, 2224년 0.0%로, 2225년 0.0%로, 2226년 0.0%로, 2227년 0.0%로, 2228년 0.0%로, 2229년 0.0%로, 2230년 0.0%로, 2231년 0.0%로, 2232년 0.0%로, 2233년 0.0%로, 2234년 0.0%로, 2235년 0.0%로, 2236년 0.0%로, 2237년 0.0%로, 2238년 0.0%로, 2239년 0.0%로, 2240년 0.0%로, 2241년 0.0%로, 2242년 0.0%로, 2243년 0.0%로, 2244년 0.0%로, 2245년 0.0%로, 2246년 0.0%로, 2247년 0.0%로, 2248년 0.0%로, 2249년 0.0%로, 2250년 0.0%로, 2251년 0.0%로, 2252년 0.0%로, 2253년 0.0%로, 2254년 0.0%로, 2255년 0.0%로, 2256년 0.0%로, 2257년 0.0%로, 2258년 0.0%로, 2259년 0.0%로, 2260년 0.0%로, 2261년 0.0%로, 2262년 0.0%로, 2263년 0.0%로, 2264년 0.0%로, 2265년 0.0%로, 2266년 0.0%로, 2267년 0.0%로, 2268년 0.0%로, 2269년 0.0%로, 2270년 0.0%로, 2271년 0.0%로, 2272년 0.0%로, 2273년 0.0%로, 2274년 0.0%로, 2275년 0.0%로, 2276년 0.0%로, 2277년 0.0%로, 2278년 0.0%로, 2279년 0.0%로, 2280년 0.0%로, 2281년 0.0%로, 2282년 0.0%로, 2283년 0.0%로, 2284년 0.0%로, 2285년 0.0%로, 2286년 0.0%로, 2287년 0.0%로, 2288년 0.0%로, 2289년 0.0%로, 2290년 0.0%로, 2291년 0.0%로, 2292년 0.0%로, 2293년 0.0%로, 2294년 0.0%로, 2295년 0.0%로, 2296년 0.0%로, 2297년 0.0%로, 2298년 0.0%로, 2299년 0.0%로, 2300년 0.0%로, 2301년 0.0%로, 2302년 0.0%로, 2303년 0.0%로, 2304년 0.0%로, 2305년 0.0%로, 2306년 0.0%로, 2307년 0.0%로, 2308년 0.0%로, 2309년 0.0%로, 2310년 0.0%로, 2311년 0.0%로, 2312년 0.0%로, 2313년 0.0%로, 2314년 0.0%로,

[illegible][illegible]

출처: 서울대학교 병원(https://www.snubh.org/service/disease/view.do?cPage=5&BNO=304&Board_ID=B004&RNUM=

- 당류 권장량에 대해

핑크 자주 먹는 과자가 있는가? 과자의 뒷면을 유심히 보면 당류 권장량 n(g), n(%)라 되어 있을 것이다. 그렇다면 하루 당류 권장량이 100g일까?

본문에서 밝혀지지 않았다. 퍼트르는 수질 오염(평균 2000ka)를 기준으로 직접적인 이 역, 우회는 전적으로 푸른 것은 (순수)물과 함께 합류하고 있다고 가정한다. 대개 물과 계곡은 순수 물을 포함 할 수 있는 하루 물의 양의 20%, 합류하는 10%에 대한 것, 총 용량은 100, 합류하는 50g이 기준이 된다. 대한민국의 강도 마찬가지로, 합류하는 것은 양적 우위를 차지하고 수질을 개선하는 데에 충분하게 문제가 되지 않는다. 즉, 물, 합류하는 50g이 기준이 된다면 2000ca, 50g 이하로 줄어든다. 본 사례의 최대 일 출물 비율도 마찬가지로 50g을 기준으로 작성되었다. 다만, 미국 상하원에서는 가능한 5%이하(상인 역 기준 25g 이하)로 제한하는 것을 권고하고 있다.

출처: 네이버 블로그(<https://m.blog.naver.com/codswallow5/221761809219>)

- 혈당을 관리하는 방법

생활 습관을 들이자

사람은 혈액 속에 함유되어 있는 크로마틴의 농도를 뜻하며, 적절한 혈당은 공복혈당 100mg/dL 미만, 비당 24시간 혈당은 140mg/dL 미만을 의미한다. 혈당은 식사 후 시간이 지남에 따라 수치가 저하되다가 다시는, 케톤체로 변을 수혈이 혈액에 나타나는 것으로 보혈을 사용해서 수치를 정확히 측정하고 정확한 혈액의 수치를 파악할 수 있다. 또한, 혈당은 반드시 수치를 실험할 때 증가하는 것은 아니다. 우리 몸은 항상 에너지 저장하는 시스템을 가지고 있다. 그래서 신체나 운동에 지친 후엔 혈당이 상승하였거나 급속히 떨어지거나, 혹은 본연의 체질에 따라 혈당이 정상에 도달하고도 변하며 다시 상승해 있다. 급속도로 저혈당이 되면, 혈당이 일정 미미한 수치를 유지하게끔 보충해 주어야 하며, 따라서 적절한 혈당관리 관리는 당뇨도와 같은 질병의 예방에 도움이 될 수 있다. 혈당이 상승하는 직접적인 원인과 원인, 건강을 위해서는 다양한 요인이나 검출을 해야 한다.

필당을 높이는 주 에너지원이 탄수화물이며, 이 탄수화물의 섭취를 줄이고 근육을 구성하고 세포의 신호전달, 성장 등에 필요한 단백질과 지방의 섭취를 늘리면 체중 조절 등에 도움 될 수 있다. 물론 탄수화물을 안 먹는 것은 매우 어려운 일이다. 따라서 식사 시 소화 속도를 조절할 수 있는 채소의 섭취를 먼저 하고 이어서 소화 흡수 속도를 늦추는 지방과 단백질을 한 후, 가공된 나중엔 탄수화물 섭취를 하게 되면, 혈당의 급격한 상승을 억제할 수 있고 인슐린의 급격한 분비 등을 조절하여 건강을 관리할 수 있다.

은근은 귀족사상을 논하는 데 도움이 되며 이를 통해 많은 음반을 먹어도 사용자는 예외를 줄일 수 있고, 또한 예외지 지평에도 도움이 된다. 특히 스프레드 원본이나 불꽃, 탄이 있는 귀족을 단련하고 예외 귀족사상 상충을 도출하고, 궁극적으로 새 귀족지도 사상을 30년에서 10년 사이 제형 될 때까지 오히려 오묘하게 예외 막을 의도적으로 추구한다. 한편, 모든 귀족은 새로운 상충을 형성할 수 있어야 하고, 또한 한 번의 예외가 지평으로 전환되는 것을 제지하는 데 도움이 된다. 개척적으로 시계 내재, 제 실수자가 되도록 형성하고 귀족사상을 새로 귀족사상을 변형 사용하는 것은 인간적 의미(비판)를 모든 인간이 지평으로 모든 귀족사상과 관련된 새로운 창작을 일기하는 것은 중요하다. 이는 실제귀족을 형성하고 귀족사상을 예외로 넘기고, 귀족사상은 모든 인간지평의 예외에도 상충하는 것을 수 있다.

p태그와 함께 페이지에서
직접 글 작성,
구글폰트를 통해 html에 전체
페이지 서체 적용

certificate(운영자 비밀번호
입력 창)로 넘어가는
하이퍼링크 연결

sugarhigh.html

슈가하이

SUGAR HIGH.

검색
게시판

수정

```
1 <!DOCTYPE html
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <link rel="preconnect" href="https://fonts.googleapis.com">
6   <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
7   <link href="https://fonts.googleapis.com/css2?family=Gowun+Batang&display=swap" rel="stylesheet">
8   <style>
9     img { display : block;
10        margin : auto; }
11    *{ font-family: 'Gowun Batang', serif; }
12  </style>
13  <a href="http://leejuyeon.site/">
14    <img src = "http://leejuyeon.site/logo" width="800" height="250">
15  </a>
16 </head>
17 <body>
18   <a href="http://leejuyeon.site/search"> <img src = "http://leejuyeon.site/srbutton" width = "80" height="40"></a>
19   <a href="http://leejuyeon.site/community"> <img src = "http://leejuyeon.site/combutton" width = "100" height="48"> </a>
20   <p style = "font-size: 35pt; text-align: left;"> - 당뇨의 위험성</p>
21   <p style = "font-size: 18pt; text-align: left;"> 당류를 과잉 섭취할 경우 기억력의 중추인 해마를 위축시켜 혈관성 치매 위험을 높
22   비만 위험도는 1.39배 높아지며, 후천성 당뇨병을 넘어 다양한 암의 위험 또한 높인다. </p>
23   <p>신체에 쌓인 당분은 비만을 유발하고, 이에 따라 인슐린 저항성이 증가하면서 뇌기능을 저하시킨다.
24   기억력의 중추인 해마(측두엽의 안쪽에 위치하면서 기억을 저장하고, 공간 개념, 감정적인 행동을 조절)를 위축시키게 되고,
25   포도당의 혈중 수치가 높은 사람이 정상인 사람에 비해 해마의 크기가 작아져 기억력이 떨어진다는 것이다.
26   또한, 과도한 설탕 섭취는 기분과 행동을 좌우하는 신경전달물질을 교란시키고 신경염증반응을 일으킴으로써 우울증 위험을 증가시킨다.
27   잘 알려져 있듯이 우울증은 치매 발생 위험을 2~3배나 증가시키게 된다.]
28   단 맛에 끌려 군것질을 자주 한다면 뇌 손상은 가중되고, 이에 따라 치매 위험도 높아지게 되는 것이다.</p>
29   <p>식사 후 30분에서 1시간 사이가 제일 혈당이 많이 오르므로 이 때 약간의 땀이 흐를 정도의 근육운
30   혈당을 소비하게 되고, 또한 남은 혈당이 지방으로 전환되는 것을 억제하는 데 도움이 된다.
31   기본적으로 짧은 시간 내에 최대 심박수에 도달할 정도의 고강도 운동을 한 후 휴식시간을 반복 수행
32   중단된 이후에도 우리 뇌가 지속적으로 운동을 하고 있다는 착각을 일으키게 하는 운동법이다. 이는
33   고혈압, 고지혈증과 같은 만성질환의 예방에도 상당한 도움을 줄 수 있다.
34   </p>
35   <p style = "color: gray; text-align: left;"> 출처: 하이닥 의료 기사(https://www.hidoc.co
36   </p>
37   <p style = "font-size: 35pt; text-align: right;"> - 사이트 내 영양정보 작성 기준</p>
38   <p style = "text-align: right;"> 본 사이트는 스타벅스, 공차, 할리스커피, 더벤티, 컴포즈커피, 매머드
39   <p style = "text-align: right;"> 신메뉴, 영양정보 변경에 대해서 주기적으로 업데이트를 하나 영양
40   <p style = "text-align: right;"> 하루 당류 총족 기준은 국제 기준인 50g을 기준으로 계산된 값입니다.
41   <p style = "text-align: right; color: green;">
42     스타벅스: 에스프레소(22ml) / 더블샷(207ml) / 험샷 코르타도(샷, 237ml) / 더 그린속 라인 , 플럼
43     <p style = "text-align: right; color: brown;">공차: ice(473ml) / hot(414ml)</p>
44     <p style = "text-align: right; color: red;">할리스 커피: 모든 메뉴 354ml</p>
45     <p style = "text-align: right; color: palevioletred;">매머드 익스프레스: ice(20온즈, 약 59
46     <p style = "text-align: right;">더벤티와 컴포즈 커피는 관련 표기 미확인 상태입니다.
47   </p>
48   <a href="http://leejuyeon.site/certification">
49     <p style = "text-align: right;"><input type = "button" id="search" value="수정"></input></p>
50   </a>
51   <p style = "font-size: 8pt; text-align: center; color: gray;">제작: 김은서</p>
52 </body>
53 </html>
```

leejuyeon.site/certificate

certificate.html

슈가하이

SUGAR HIGH.

비밀번호 입력 :

인증

인증

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Gowun+Batang&display=block" rel="stylesheet">
    <style>
      img { display : block;
        margin : auto; }
      *{ font-family: 'Gowun Batang', serif; }
    </style>
    <a href="http://leejuyeon.site/">
      <img src = "http://leejuyeon.site/logo" width="800" height="250">
    </a>
  </head>
  <body>
    <form>
      <div>
        <p style = "text-align: center;">
          비밀번호 입력 : <input id="pw" type="password" name="pw" size="50">
          <input type = "submit" id="login" value="인증"></input></p>
        </div>
      </form>
      <script>
        document.getElementById("login").onclick = function() {
          const url = 'http://leejuyeon.site/root';
          fetch(url)
            .then((response) => response.json())
            .then((data)=>{
              data.forEach(element => {
                if(document.getElementById("pw").value == element.passwd) {
                  // success
                } else { alert("비밀번호 오류"); }
              });
            });
        }
      </script>
    </body>
  </html>
```


app.js

```
app.get('/root', (req, res) => {  
  let sql = "SELECT * FROM account";  
  connection.query(sql, function (err, results, fields) {  
    if (err) {  
      console.log(err);  
    }  
    console.log('ok');  
    list = JSON.stringify(results);  
    res.send(list);  
  });  
})
```

```
[{"uid":1,"name":"root","passwd":'
```

슈가하이

SUGAR HIGH.

- 커뮤니티 글 삭제

요청사항이 구현완료된 글은 삭제처리하여 게시판을 깔끔하게 유지합니다.

삭제할 글(code) 입력 :

- 내용 삭제

필드 목록 / 값은 아래 내용확인을 통해 자세한 사항 확인 바랍니다.

삭제후 혼동을 방지하기 위해 꼭 아래 mid 최종값을 업데이트 해주시기 바랍니다.

삭제할 행(mid) 입력 :

- 내용 수정

수정은 하나의 필드씩만 가능합니다.

필드 목록 / 값은 아래 내용확인을 통해 자세한 사항 확인 바랍니다.

percentage의 계산은 아래 내용추가란을 통해 쉽게 계산할 수 있습니다.

중복방지를 위해 mid값은 수정할 수 없습니다. 데이터베이스를 통해 직접 수정 바랍니다.

수정할 필드 : / 수정할 필드 값: / 조건(mid) 입력 :

- 내용 추가

스타벅스(1) : 107 / 공차(2) : 75 / 할리스(3) : 61 / 더벤티(4) : 109 / 컴포즈(5) : 59 / 매머드익스프레스(6) : 121 (2023.05.10 기준)

percentage 계산:

mid	name	sugar	kcal	percentage	color	brand	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="추가"/>
<input type="button" value="내용 확인"/>							
<input type="text" value="mid"/>	<input type="text" value="name"/>	<input type="text" value="sugar"/>	<input type="text" value="kcal"/>	<input type="text" value="percentage"/>	<input type="text" value="color"/>	<input type="text" value="brand"/>	

표 개체를 통해 정렬된
텍스트박스 생성

modify.html

- 커뮤니티 글 삭제

요청사항이 구현완료된 글은 삭제처리하여 게시판을 깔끔하게 유지합니다.

삭제할 글(code) 입력 :

```
<p style = "font-size: 35pt; text-align: left;"> - 커뮤니티 글 삭제</p>
<p> 요청사항이 구현완료된 글은 삭제처리하여 게시판을 깔끔하게 유지합니다. </p>
<p> </p>
<form action="/comdelete" method="post">
  <p>
    삭제할 글(code) 입력 : <input type="text" name="where" size="10">
    <input type = "submit" id="delete" value="삭제"></input>
  </p>
</form>
```

- 내용 삭제

필드 목록 / 값은 아래 내용확인을 통해 자세한 사항 확인 바랍니다.

삭제후 혼동을 방지하기 위해 꼭 아래 mid 최종값을 업데이트 해주시기 바랍니다.

삭제할 행(mid) 입력 :

```
<p style = "font-size: 35pt; text-align: left;"> - 내용 삭제</p>
<p> 필드 목록 / 값은 아래 내용확인을 통해 자세한 사항 확인 바랍니다.</p>
<p> 삭제후 혼동을 방지하기 위해 꼭 아래 mid 최종값을 업데이트 해주시기 바랍니다.</p>
<form action="/dbdelete" method="post">
  <p>
    삭제할 행(mid) 입력 : <input type="text" name="where" size="10">
    <input type = "submit" id="delete" value="삭제"></input>
  </p>
</form>
```

- 내용 수정

수정은 하나의 필드씩만 가능합니다.

필드 목록 / 값은 아래 내용확인을 통해 자세한 사항 확인 바랍니다.

percentage의 계산은 아래 내용추가란을 통해 쉽게 계산할 수 있습니다.

중복방지를 위해 mid값은 수정할 수 없습니다. 데이터베이스를 통해 직접 수정 바랍니다.

수정할 필드 : / 수정할 필드 값: / 조건(mid) 입력 :

```
<p style = "font-size: 35pt; text-align: left;"> - 내용 수정</p>
<p> 수정은 하나의 필드씩만 가능합니다.</p>
<p> 필드 목록 / 값은 아래 내용확인을 통해 자세한 사항 확인 바랍니다.</p>
<p> percentage의 계산은 아래 내용추가란을 통해 쉽게 계산할 수 있습니다.</p>
<p> 중복방지를 위해 mid값은 수정할 수 없습니다. 데이터베이스를 통해 직접 수정 바랍니다.</p>
<form action="/dbupdate" method="post">
  <p>
    수정할 필드 :
    <select name="field" id="field">
      <option value="name">name</option>
      <option value="sugar">sugar</option>
      <option value="kcal">kcal</option>
      <option value="percentage">percentage</option>
      <option value="color">color</option>
      <option value="brand">brand</option>
    </select>
    / 수정할 필드 값: <input type="text" name="field2" size="30">
    / 조건(mid) 입력 : <input type="text" name="where" size="10">
    <input type = "submit" id="update" value="수정"></input>
  </p>
</form>
```

아무 값도 들어오지 않은 채
버튼이 눌러 데이터가
전송되었다면 sql문을
실행하지 않음

```
app.post('/comdelete', (req, res) => {  
  var body = '';  
  var flag = false;  
  req.on('data', function(data) {body=body+data;});  
  req.on('end', function() {  
    var post = qs.parse(body);  
    const obj=JSON.parse(JSON.stringify(post));  
    var keys=Object.keys(obj);  
    let sql="DELETE FROM community WHERE code=?"  
    var params = [];  
    params.forEach(element => { if(element==null || element==NaN) flag=true; });  
    if(flag!=true){  
      connection.query(sql, params, function (err, rows, fields) {  
        if (err) {  
          console.log(err);  
        }  
        console.log("ok");  
      });  
      res.writeHead(302, {Location: `/modify`});  
      res.end();  
    }  
    else {  
      console.log("삭제할 수 없습니다.")  
      res.writeHead(302, {Location: `/modify`});  
      res.end();  
    }  
  })  
})
```

페이지 되돌아가기

```
app.post('/dbdelete', (req, res) => {  
  var body='';  
  var flag = false;  
  req.on('data', function(data) {body=body+data;});  
  req.on('end', function() {  
    var post = qs.parse(body);  
    const obj=JSON.parse(JSON.stringify(post));  
    var keys=Object.keys(obj);  
    let sql="DELETE FROM menu WHERE mid=?"  
    var params = [Number(obj[keys[0]])];  
    params.forEach(element => { if(element==null || element==NaN) flag=true; });  
    if(flag!=true){  
      connection.query(sql, params, function (err, rows, fields) {  
        if (err) {  
          console.log(err);  
        }  
        console.log("ok");  
      });  
      res.writeHead(302, {Location : `/modify`});  
      res.end();}  
    else {  
      console.log("삭제할 수 없습니다.")  
      res.writeHead(302, {Location : `/modify`});  
      res.end(); }  
  })  
})
```

```

app.post('/dbupdate', (req, res) => {
  var body='';
  var flag = false;
  req.on('data', function(data) {body=body+data;});
  req.on('end', function() {
    var post = qs.parse(body);
    const obj=JSON.parse(JSON.stringify(post));
    var keys=Object.keys(obj);
    let sql;
    var params = [obj[keys[0]], obj[keys[1]], obj[keys[2]]];
    params.forEach(element => { if(element==null || element==NaN) flag=true; });
    if(obj[keys[0]]=="name") {sql="UPDATE menu SET name = ? WHERE mid=?"; params = [obj[keys[1]], Number(obj[keys[2]])];}
    else if(obj[keys[0]]=="sugar") {sql="UPDATE menu SET sugar = ? WHERE mid=?"; params = [Number(obj[keys[1]]), Number(obj[keys[2]])];}
    else if(obj[keys[0]]=="kcal") {sql="UPDATE menu SET kcal = ? WHERE mid=?"; params = [Number(obj[keys[1]]), Number(obj[keys[2]])];}
    else if(obj[keys[0]]=="percentage") {sql="UPDATE menu SET percentage = ? WHERE mid=?"; params = [Number(obj[keys[1]]), Number(obj[keys[2]])];}
    else if(obj[keys[0]]=="color") {sql="UPDATE menu SET color = ? WHERE mid=?"; params = [obj[keys[1]], Number(obj[keys[2]])];}
    else if(obj[keys[0]]=="brand") {sql="UPDATE menu SET brand = ? WHERE mid=?"; params = [obj[keys[1]], Number(obj[keys[2]])];}
    else { flag=true; }
    if(flag!=true){
      connection.query(sql, params, function (err, rows, fields) {
        if (err) {
          console.log(err);
        }
        console.log("ok");
      });
      res.writeHead(302, {Location: `/modify`});
      res.end();}
    else {
      console.log("null값은 넣을 수 없습니다.")
      res.writeHead(302, {Location: `/modify`});
      res.end(); }
  })
})

```

percentage 계산:

```

61 <p style = "font-size: 35pt; text-align: left;"> - 내용 추가</p>
62 <p> 스타벅스(1) : 107 / 공차(2) : 75 / 할리스(3) : 61 / 더벤티(4) : 109 / 컴포즈(5) : 59 / 매머드익스프레스(6) : 121 (2023.05.10 기준) </p>
63 <p> percentage 계산: <input type="text" id="res" size="10"> <input type = "button" id="calculate" value="계산"></input>
64 <script>
65     document.getElementById("calculate").onclick = function(){
66         var input = parseInt(document.getElementById("res").value);
67         const output = (input/50)*100;
68         alert(output); }
69 </script>
70 </p>
71 <div>
72 <table>
73 <thead>
74 <tr>
75 <th>mid</th>
76 <th>name</th>
77 <th>sugar</th>
78 <th>kcal</th>
79 <th>percentage</th>
80 <th>color</th>
81 <th>brand</th>
82 </tr>
83 </thead>
84 <form action="/dbinsert" method="post">
85 <tr>
86 <td><p><input type="number" id="mid" name="mid"></p></td>
87 <td><p><input type="text" id="name" name="name" autocomplete="off"></p></td>
88 <td><p><input type="number" id="sugar" name="sugar" step="0.01"></p></td>
89 <td><p><input type="number" id="kcal" name="kcal" step="0.01"></p></td>
90 <td><p><input type="number" id="percentage" name="percentage" step="0.01"></p></td>
91 <td><p><input type="text" id="color" name="color" autocomplete="off"></p></td>
92 <td><p><input type="text" id="brand" name="brand" autocomplete="off"></p></td>
93 <td><p><input type="submit" id="plus" name ="plus" style="width: 50pt; height: 20pt;" value="추가"></p></td>
94 </tr>
95 </form>
96 </table>
97 </div>

```

- 내용 추가

스타벅스(1) : 107 / 공차(2) : 75 / 할리스(3) : 61 / 더벤티(4) : 109 / 컴포즈(5) : 59 / 매머드익스프레스(6) : 121 (2023.05.10 기준)

percentage 계산:

mid	name	sugar	kcal	percentage	color	brand
-----	------	-------	------	------------	-------	-------

autocomplete: 자동완성 사용여부
step: 소수점

```
app.post('/dbinsert', (req, res) => {  
  var body = '';  
  var flag = false;  
  req.on('data', function(data) {body=body+data;});  
  req.on('end', function() {  
    var post = qs.parse(body);  
    const obj=JSON.parse(JSON.stringify(post));  
    var keys=Object.keys(obj);  
    let sql="INSERT INTO menu VALUES(?, ?, ?, ?, ?, ?, ?)"  
    var params = [Number(obj[keys[0]]), obj[keys[1]], Number(obj[keys[2]]), Number(obj[keys[3]]), Number(obj[keys[4]]), obj[keys[5]], obj[keys[6]]];  
    params.forEach(element => {  
      if(element==null || element==NaN) flag=true;  
    });  
    if(flag!=true){  
      connection.query(sql, params, function (err, rows, fields) {  
        if (err) {  
          console.log(err);  
        }  
        console.log("ok");  
      });  
      res.writeHead(302, {Location : `/modify`});  
      res.end();}  
    else {  
      console.log("null값은 넣을 수 없습니다.")  
      res.writeHead(302, {Location : `/modify`});  
      res.end(); }  
  })  
})
```



```
<button id="confirm">내용 확인</button>
<table id="result" border="1">
  <thead>
    <tr>
      <th>mid</th>
      <th>name</th>
      <th>sugar</th>
      <th>kcal</th>
      <th>percentage</th>
      <th>color</th>
      <th>brand</th>
    </tr>
  </thead>
```

```
<tbody id = "this">
```

```
<script>
```

```
const show = document.getElementById("confirm");
```

```
show.onclick = function() {
```

```
while(document.getElementById("result").rows.length>1){ document.getElementById("result").deleteRow(1); }
```

```
const url = "http://leejuyeon.site/menu ;
```

```
fetch(url)
```

```
.then((response) => response.json())
```

```
.then((data)=>{
```

```
const tbody = document.getElementById("this");
```

```
data.forEach(element => {
```

```
const tr = document.createElement('tr');
```

```
tr.innerHTML = `<th>${element.mid}</th> <th>${element.name}</th> <th>${element.sugar}</th> <th>${element.kcal}</th> <th>${element.percentage}</th> <th>${element.color}</th> <th>${element.brand}</th>`;
```

```
tbody.appendChild(tr);
```

```
});
```

```
});
```

```
}
```

```
</script>
```

내용 확인

mid	name	sugar	kcal	percentage	color	brand
-----	------	-------	------	------------	-------	-------

버튼을 누를 때마다 데이터가 추가되는 것을 방지하기 위해, 만약 버튼을 눌렀을 때 표에 데이터가 있다면 기존데이터를 모두 삭제함

표의 첫번째 행(thead) 아래부터 sql문의 결과를 받아와 append함

app.js

```
app.get('/menu', (req, res) => {
  let sql = "SELECT * FROM menu";
  connection.query(sql, function (err, results, fields) {
    if (err) {
      console.log(err);
    }
    console.log('ok');
    list = JSON.stringify(results);
    res.send(list);
  });
});
```

leejuyeon.site/modify

내용 확인						
mid	name	sugar	kcal	percentage	color	brand
1001	리저브나이트로	0	5	0	brown	스타벅스
1002	나이트로 바닐라 크림	10	80	20	brown	스타벅스
1003	나이트로 콜드브루	0	5	0	brown	스타벅스
1004	돌체 콜드브루	29	265	58	brown	스타벅스
1005	리저브 콜드브루	0	5	0	black	스타벅스
1006	민트 콜드브루	23	100	46	brown	스타벅스
1007	바닐라 크림 콜드브루	11	125	22	brown	스타벅스
1008	시그니처 더 블랙 콜드브루	0	25	0	black	스타벅스
1009	콜드브루	0	5	0	black	스타벅스

```
[{"mid":1001,"name":"리저브나이트로","sugar":0,"kcal":5,"percentage":0,"color":"brown","brand":"스타벅스"}, {"mid":1002,"name":"나이트로 바닐라 크림","sugar":10,"kcal":80,"percentage":20,"color":"brown","brand":"스타벅스"}, {"mid":1003,"name":"나이트로 콜드브루","sugar":0,"kcal":5,"percentage":0,"color":"brown","brand":"스타벅스"}, {"mid":1004,"name":"돌체 콜드브루","sugar":29,"kcal":265,"percentage":58,"color":"brown","brand":"스타벅스"}, {"mid":1005,"name":"리저브 콜드브루","sugar":0,"kcal":5,"percentage":0,"color":"black","brand":"스타벅스"}, {"mid":1006,"name":"민트 콜드브루","sugar":23,"kcal":100,"percentage":46,"color":"brown","brand":"스타벅스"}, {"mid":1007,"name":"바닐라 크림 콜드브루","sugar":11,"kcal":125,"percentage":22,"color":"brown","brand":"스타벅스"}, {"mid":1008,"name":"시그니처 더 블랙 콜드브루","sugar":0,"kcal":25,"percentage":0,"color":"black","brand":"스타벅스"}, {"mid":1009,"name":"콜드브루","sugar":0,"kcal":5,"percentage":0,"color":"black","brand":"스타벅스"}, {"mid":1011,"name":"콜드브루 오프라떼","sugar":14,"kcal":120,"percentage":28,"color":"brown","brand":"스타벅스"}, {"mid":1012,"name":"콜드브루 트","sugar":18,"kcal":225,"percentage":36,"color":"brown","brand":"스타벅스"}, {"mid":1013,"name":"콜드브루 오프라떼","sugar":41,"kcal":345,"percentage":82,"color":"brown","brand":"스타벅스"}, {"mid":1016,"name":"에스프레소 콘파나","sugar":1,"kcal":30,"percentage":2,"color":"brown","brand":"스타벅스"}, {"mid":1017,"name":"에스프레소","sugar":0,"kcal":10,"percentage":0,"color":"brown","brand":"스타벅스"}, {"mid":1019,"name":"카페 아메리카노","sugar":0,"kcal":10,"percentage":0,"color":"brown","brand":"스타벅스"}]
```

슈가 하 이

SUGAR HIGH.

검 색
게시판

이름

검색

! 검색 관련 팁 !

1. 이름 / 브랜드 / 당류 / 칼로리 / 하루 당류 권장 비율 / 색상에 따라 검색 가능합니다.

2. 영양정보 기준은 다음과 같습니다.

하루 최대 당류 총족 비율은 국제 기준인 50g을 기준으로 계산된 값입니다.

스타벅스: 에스프레소(22ml) / 더블샷(207ml) / 텀샷 코르타도(샷, 237ml) / 더 그린썬 라인 , 플럼 섀트(그란데, 473ml) / 이 외의 메뉴는 모두 톨(335ml)사이즈

공차: ice(473ml) / hot(414ml)

할리스 커피: 모든 메뉴 354ml

매머드 익스프레스: ice(20온즈, 약 590ml) / hot(16온즈, 약 473ml) / 매머드사이즈(32온즈, 약 946ml)

더벤터와 컴포즈 커피는 관련 표기 미확인 상태입니다.

3. 색상표기는 black, brown, ivory, red, yellow, pink, green, purple, orange, blue 로만 표기되어있습니다.

4. 당류, 칼로리, 하루 당류 권장 비율 카테고리는 숫자로만 검색이 가능하며, 검색값 이하의 메뉴를 출력합니다.

ex. '34.5' : 당류 34.5g이하의 메뉴 출력 | '100' : 하루 당류 권장 비율 100% 이하의 메뉴 출력

5. 음료 커스텀 시, 사이트에 표기된 값과 실제 값에 차이가 있을 수 있습니다.

i. 스타벅스는 톨 사이즈(335ml)를 기준으로 표기된 값입니다. 사이즈를 한단계 높였을 때, 일반적으로 칼로리는 10% ~ 최대 30% 증가 합니다.

ii. 공차는 당도 0%일때를 기준으로 표기된 값입니다. 당도가 20% 올랐을 때 일반적으로 당류는 3~5g 정도가 추가됩니다.

6. 오타 및 관련 내용 오류에 관해서는 게시판에 글을 작성하여 수정을 요청해주세요.

브랜드 이름 당류 칼로리 하루 최대 당류 총족 비율(%)

searchpage.html

이름

검색

```
<form action="/src" method="post">
  <p style="text-align: center; font-size: 30pt">
    <select name="field" id="field" style="font-size: 30pt;">
      <option value="name">이름</option>
      <option value="sugar">당류</option>
      <option value="kcal">칼로리</option>
      <option value="percentage">당류 비율</option>
      <option value="color">색깔</option>
      <option value="brand">브랜드</option>
    </select>
    <input type="text" name="searchbox" id="searchbox" size="25" style="font-size: 30pt;">
    <input type="submit" id="search" value="검색" style="font-size: 30pt;"></input>
  </p>
</form>
```

```

app.post('/src', (req, res) => {
  var list;
  var body='';
  var flag = false;
  req.on('data', function(data) {body=body+data;});
  req.on('end', function() {
    var post = qs.parse(body);
    const obj=JSON.parse(JSON.stringify(post));
    var keys=Object.keys(obj);
    var params = [obj[keys[0]], obj[keys[1]]];
    params.forEach(element => { if(element==null || element==NaN) {flag=true; } });
    if(obj[keys[0]]=="name") {sqls="SELECT * FROM menu WHERE name LIKE ?"; param = '%' + [obj[keys[1]]] + '%';}
    else if(obj[keys[0]]=="sugar") {sqls="SELECT * FROM menu WHERE sugar<=?"; param = [Number(obj[keys[1]])]; }
    else if(obj[keys[0]]=="kcal") {sqls="SELECT * FROM menu WHERE kcal<=?"; param = [Number(obj[keys[1]])]; }
    else if(obj[keys[0]]=="percentage") {sqls="SELECT * FROM menu WHERE percentage<=?"; param = [Number(obj[keys[1]])]; }
    else if(obj[keys[0]]=="color") {sqls="SELECT * FROM menu WHERE color LIKE ?"; param = '%' + [obj[keys[1]]] + '%';}
    else if(obj[keys[0]]=="brand") {sqls="SELECT * FROM menu WHERE brand LIKE ?"; param = '%' + [obj[keys[1]]] + '%';}
    else { flag=true; }
    if(flag!=true){
      connection.query(sqls, param, function (err, results, fields) {
        if (err) {
          console.log(err);
        }
        list = JSON.stringify(results);
        console.log("ok");
      });
      res.writeHead(302, {Location : `/search`});
      res.end();}
    else {
      console.log("값이 잘못되었습니다.")
      res.writeHead(302, {Location : `/search`});
      res.end(); }
  })
})

```

검색 지원은 이름, 당류의 양, 열량, 비율, 색깔,
브랜드만 지원함 (각 메뉴의 고유코드는 사용자가 알
필요가 없음)

sql문 저장 후, '%'를 검색값(파라미터)
앞뒤로 추가하여 파라미터가 들어간
모든 값을 검색할 수 있도록 함

searchpage.html

브랜드	이름	당류	칼로리	하루 최대 당류 충족 비율(%)
-----	----	----	-----	-------------------

```
<thead>
  <table id="result" border="1" align = "center">
    <tr>
      <th>브랜드</th>
      <th>이름</th>
      <th>당류</th>
      <th>칼로리</th>
      <th>하루 최대 당류 충족 비율(%)</th>
    </tr>
  </thead>
<tbody>
<script>
  const url = 'http://leejuyeon.site/buffer';
  fetch(url)
    .then((response) => response.json())
    .then((data)=>{
      const tbody = document.querySelector('tbody');
      data.forEach(element => {
        const tr = document.createElement('tr');
        tr.innerHTML = `<th>${element.brand}</th> <th>${element.name}</th> <th>${element.sugar}</th> <th>${element.kcal}</th> <th>${element.
        tbody.appendChild(tr);
      });
    });
</script>
</tbody>
</table>
```

저장한 sql문 재실행, 결과를 보내어 출력할 수 있도록 함

```
app.get('/buffer', (req, res) => {  
  var sql = sqls;  
  var list;  
  connection.query(sql, param, function (err, results, fields) {  
    if (err) {  
      console.log(err);  
    }  
    console.log('ok');  
    list = JSON.stringify(results);  
    res.send(list);  
  });  
})
```

슈가 하 이

SUGAR HIGH.

검 색 게시판

구현 완료된 요청사항은 삭제됩니다.

code	name	request
1	akira	스타벅스 신메뉴도 추가해주세요
2	sendoh	메가커피도 넣어주세요
3	maki	공차 타로버블티(필) 정보 재확인 부탁드립니다

각 칸당 여백을 주지 않은 표

글쓰기

하이퍼링크


```
<p style="text-align: center;"> 구현 완료된 요청사항은 삭제됩니다. </p>
<thead>
  <table align = "center">
    <tr>
      <th>code</th>
      <th>name</th>
      <th>request</th>
    </tr>
  </thead>
</table>
<script>
  const url = 'http://leejuyeon.site/request';
  fetch(url)
    .then((response) => response.json())
    .then((data)=>{
      const tbody = document.querySelector('tbody');
      data.forEach(element => {
        const tr = document.createElement('tr');
        tr.innerHTML = `<th>${element.code}</th> <th>${element.nickname}</th> <th>${element.request}</th>`;
        tbody.appendChild(tr);
      });
    });
</script>
<a href="http://leejuyeon.site/writer">
  <p style = "text-align: right"><input type = "button" id="search" value="글쓰기"></p>
</a>
```

글쓰기

```
app.get('/request', (req, res) => {  
  let sql = "SELECT * FROM community";  
  var list;  
  connection.query(sql, function (err, results, fields) {  
    if (err) {  
      console.log(err);  
    }  
    console.log('ok');  
    list = JSON.stringify(results);  
    res.send(list);  
  });  
})
```

```
[{"code":1,"nickname":"akira","request":"스타벅스 신메뉴도 추가해주세요"}, {"code":2,"nickname":"sendoh",  
인 부탁드립니다"}]
```

슈가하이

SUGAR HIGH.

검색
게시판

code :

이름 :

정보 요청 / 수정 등 요청 내용을 입력해주세요.

code : 0은 코드로 사용 불가

중복확인

```
<form action = "/requestconfirm" method="post">
```

```
<p style="text-align: center;">
```

```
code : <input type="number" id="code" name="코드" size="10" min="1" placeholder="0은 코드로 사용 불가"
```

```
onfocus="this.placeholder=''" onblur="this.placeholder='0은 코드로 사용 불가'">
```

```
<input type="button" id="duplicate" name="중복확인" value="중복확인">
```

```
<script>
```

```
let flag=true;
```

```
document.getElementById("duplicate").onclick = function() {
```

```
const url = 'http://leejuyeon.site/request';
```

```
fetch(url)
```

```
.then((response) => response.json())
```

```
.then((data)=>{
```

```
data.forEach(element => {
```

```
if(element.code===Number(document.getElementById('code').value)) { flag=false; }
```

```
if(document.getElementById('code').value=="") { flag=false; }
```

```
});
```

```
if(flag==true) { alert("사용가능한 코드입니다.");}
```

```
else{ alert("사용할 수 없는 코드입니다."); flag=true; }
```

```
}}
```

```
</script>
```

```
</p>
```

```
<p style="text-align: center;">
```

```
이름 : <input type="text" id="nickname" name="이름" size="30" placeholder="코드 중복확인을 눌러주세요!"
```

```
onfocus="this.placeholder=''" onblur="this.placeholder='코드 중복확인을 눌러주세요!'">
```

```
</p>
```

```
<p style="text-align: center;">
```

```
<textarea name="내용" id="request" rows="10" cols="37" placeholder="정보 요청 / 수정 등 요청 내용을 입력해주세요."
```

```
onfocus="this.placeholder=''" onblur="this.placeholder='정보 요청 / 수정 등 요청 내용을 입력해주세요.'"></textarea>
```

```
</p>
```

```
<p style = "text-align: center;"><input type = "submit" id="write" value="완료"></input></p>
```

```
</form>
```

```
</body>
```

placeholder / onfocus /
onblur를 통해 누르면 사라지는
텍스트 제작

```
app.post('/requestconfirm', (req, res) => {
  var flag=false;
  var body='';
  req.on('data', function(data) {body=body+data;});
  req.on('end', function() {
    var post = qs.parse(body);
    const obj=JSON.parse(JSON.stringify(post));
    var keys=Object.keys(obj);
    let sql="INSERT INTO community(code, nickname, request) VALUES(?, ?, ?)"
    var params = [Number(obj[keys[0]]), obj[keys[1]], obj[keys[2]]];
    if(obj[keys[0]]==0) {flag=true;}
    if(flag==false){
      connection.query(sql, params, function (err, rows, fields) {
        if (err) {
          console.log(err);
        }
        console.log('confirm');
      });
    }
    res.writeHead(302, {Location : `~/community`});
    res.end();
  })
})
```

코드가 입력되지 않았거나, 0이
입력되었다면 sql문을 실행하지
않음



homepage



<http://leejuyeon.site/>