

바쁜 현대인들을 위한

What 2 Eat

: 스위치온 다이어트 가이드와 영양소 체크 서비스

스위치온 다이어트



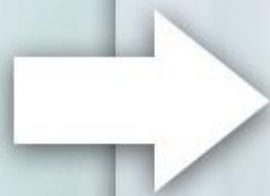
스위치온 다이어트

강북삼성병원 가정의학과 전문의 박용우 박사가
33년 비만치료를 해오면서 정리한 프로그램



105.4kg

지방간
인슐린 저항성
전당뇨
고혈압 위험
고지혈증 위험



96.8kg

대사
이상
탈출

스위치온 다이어트 챌린지
단 4주만에 바뀐 변화

기본규칙

- 1. 하루에 물 8컵 이상(2리터)을 섭취한다.
- 2. 수면시간은 최소 6시간 이상을 확보하고, 자정~오전 4시는 필수로 포함한다.
- 3. 필요한 영양제를 꼭 섭취한다.
- 4. 간헐적 단식을 실시한다.(기본 14시간 단식-10시간 식사)
- 5. 저녁식사는 취침 4시간 이전에 끝낸다.
- 6. 주 4회 이상 규칙적으로, 고강도 운동을 실시한다.
- 7. 오래 가만히 앉아있지 않는다.

1단계: 1~3일차 - 지방 대사 스위치 켜기

1~3일차는 장에 휴식을 주고, 생체리듬을 개선하는 것을 목표로 한다.

하루 4번(아침, 점심, 오후간식, 저녁) 단백질 음료만 섭취한다.(물 또는 무가당 두유와 섭취)

심한 허기로 힘이 들면 녹황색 채소, 두부, 코코넛 오일, 올리브 오일, 무가당 플레인요거트 등은 허용한다.

아침공복에 유산균을 챙겨먹어야 한다.

하루 60분 이상 걸을 수 있도록 노력한다.

* 기존의 몸 상태가 좋지 않았거나 카페인 중독이었을 경우 심한 무기력감, 두통, 현기증이 나타날 수 있다. 4일차 점심을 먹으면 대부분 사라지나, 극심한 어지럼증, 구토증세 등으로 일상생활이 어려울 정도라면 이 구간은 생략하고 4~7일차로 넘어가도 된다. 몸상태를 보고 결정해야 한다.

2단계: 4~7일차 - 렙틴 저항성과 지방간 개선

허용식품에 콩류, 견과류, 약간의 김치, 오전 중 블랙커피 1잔, 우유 2잔과 짜지 않은 천연 치즈가 추가된다.

하루 2번(아침, 오후간식) 단백질 음료를 섭취한다.

점심식사는 저탄수화물 식단으로, 4~7일차와 동일하게 실시한다.

저녁식사는 무탄수화물 식단으로, 단백질이 충분한 식사로 등푸른 생선, 가급적 굵지 않고 끓이거나 삶은 고기류(수육, 샤브샤브 등) 등을 배부르게 먹는다.

주 1회 24시간 단식을 실시한다. 자정부터 24시간 굶는 것이 아니라 이른 저녁식사 후 익일 오후 간식까지 단식 후 저녁식사를 정상적으로 먹는 것이다.

단식 하는 날은 고강도 운동은 피하되, 적절한 수준의 운동을 실시하면 효과가 더 좋다. 단식 후에는 무탄수 식사를 배부르게 먹는다.

3단계: 2주차 - 간헐적 단식과 인슐린 저항성 개선

허용식품에 콩류, 견과류, 약간의 김치, 오전 중 블랙커피 1잔, 우유 2잔과 짜지 않은 천연 치즈가 추가된다.

하루 2번(아침, 오후간식) 단백질 음료를 섭취한다.

점심식사는 저탄수화물 식단으로, 4~7일차와 동일하게 실시한다.

저녁식사는 무탄수화물 식단으로, 단백질이 충분한 식사로 등푸른 생선, 가급적 굵지 않고 끓이거나 삶은 고기류(수육, 샤브샤브 등) 등을 배부르게 먹는다.

주 1회 24시간 단식을 실시한다. 자정부터 24시간 굶는 것이 아니라 이른 저녁식사 후 익일 오후 간식까지 단식 후 저녁식사를 정상적으로 먹는 것이다.

단식 하는 날은 고강도 운동은 피하되, 적절한 수준의 운동을 실시하면 효과가 더 좋다. 단식 후에는 무탄수 식사를 배부르게 먹는다.

4단계: 3주차 - 대사유연성 회복과 신진대사 최적화

허용식품에 단호박, 토마토, 방울토마토, 밤, 바나나, 베리류 과일 등이 추가된다. 양 조절은 필수이다.

하루 2번(아침, 오후간식) 단백질 음료를 섭취한다.

점심식사는 저탄수화물 식단, 저녁식사는 무탄수화물 식단으로 2주차와 동일하게 실시한다.

고강도 운동을 했다면 바나나와 고구마 등의 탄수화물 간식도 가능하나, 추천하지 않는다.

주 2회 24시간 단식을 실시한다, 단, 이를 연속으로 수행하지 않는다.

단식 후 고단백 식사를 충분히 해주어야 근손실이 방지된다. 3주차를 종료하고 나서 근육량이 다이어트 시작일만큼 회복되었다면 4주차로 진행하고, 아니라면 다시 수행한다.

5단계: 4주차 - 업그레이드! 체지방 감량 극대화하기

허용식품에 과일이 추가된다(하루 1개로 제한) 하루 2번(아침, 오후간식) 단백질 음료를 섭취한다.

점심식사와 저녁식사 모두 저탄수화물 식단을 한다. 주 3회 24시간 단식을 실시한다. 단, 연말아 수행하지 않는다.

금기 식품

- 1. 설탕, 액상과당
- 2. 밀가루 음식
- 3. 술
- 4. 포화지방이 많은 음식
- 5. 트랜스지방이 많은 음식
- 6. 카페인이 함유된 음식 당초 커피도 금지식품이었으나, 직장인 환자들의 강력한 반발(?)로 2주차부터 오전 중 블랙커피는 가능하다.
- 7. 대체당(제로음료 등): 식사 시 약간의 간을 위한 소량의 사용은 괜찮으나, 제로콜라 등 강한 단맛을 주는 것은 피한다. 4주간 단맛 자체를 멀리하도록 한다.

* 1단계 : 아침, 점심, 저녁 간식, 저녁은 단백질 음료로 대체하며, 혼용 음료는 물이나 무가당 두유 제외 다른 음료 대체 안됨.심한 허기로 힘이 들면 녹황색 채소, 두부, 코코넛 오일, 올리브 오일, 무가당 플레인요거트 등은 허용.

* 2단계 : 하루 3번(아침, 오후간식, 저녁) 단백질 음료를 섭취, 점심식사는 저탄수화물 식단으로, 잡곡밥 반공기와 두부, 생선, 회, 해산물, 참치, 닭고기(껍질제외), 달걀, 버섯, 해조류 등은 허용하며, 아침 저녁으로 유산균, 종합비타민, 오메가3, 비타민C, CoQ10 등의 영양제를 섭취.

* 3단계 : 허용식품에 콩류, 견과류, 약간의 김치, 오전 중 블랙커피 1잔, 우유 2잔과 짜지 않은 천연 치즈가 추가, 하루 2번(아침, 오후간식) 단백질 음료를 섭취. 점심식사는 저탄수화물 식단으로, 4~7일차와 동일하게 실시. 저녁식사는 무탄수화물 식단으로, 단백질이 충분한 식사로 등푸른 생선, 가급적 굵지 않고 끓이거나 삶은 고기류(수육, 샤브샤브 등) 등을 허기가 느껴지지 않을 정도로 섭취.

* 4단계 : 허용식품에 단호박, 토마토, 방울토마토, 밤, 바나나, 베리류 과일 등이 추가되며. 양 조절은 필수. 하루 2번(아침, 오후간식) 단백질 음료를 섭취. 점심식사는 저탄수화물 식단, 저녁식사는 무탄수화물 식단으로 2주차와 동일하게 실시. 고강도 운동을 했다면 바나나와 고구마 등의 탄수화물 간식도 가능하나, 추천 X.

* 5단계 : 허용식품에 과일이 추가됨(하루 1개로 제한) 하루 2번(아침, 오후간식) 단백질 음료를 섭취. 점심식사와 저녁식사 모두 저탄수화물 식단.

1단계(1~3일차)

단백질셰이크, 신바이오텍스 혹은 프로바이오텍스, 양파, 마늘, 고춧가루, 식초, 후추, 강황, 허브, 양배추, 무, 당근, 오이, 브로콜리, 파프리카, 아보카도, 코코넛오일, 올리브오일, 아보카도오일, 냉압착 들기름, 녹차, 허브티, 무가당 플레인요거트, 두부, 연두부, 무가당 두유

2단계(4~7일차)

1단계 허용식품에 아래 추가

현미잡곡밥 2/3공기 혹은 흰쌀밥 반 공기, 해조류(미역, 다시마, 톳), 버섯류, 와사비, 저염간장(약간), 두부쌈장, 김치(약간), 달걀, 생선, 생선회, 해산물(굴, 조개, 새우, 게, 가재, 오징어, 낙지, 문어), 닭고기(껍질 벗긴 속살), 삶은 돼지고기 살코기(수육), 소고기 살코기 샤브샤브

3단계(2주차)

2단계 허용식품에 아래 추가

우유(하루 두 잔 이하), 짜지 않은 천연치즈, 퀴노아, 콩류(검은콩, 병아리콩, 완두콩, 렌틸콩 등), 견과류(한 줌), 블랙커피(오전 중 한 잔)

4단계(3주차) 및 5단계(4주차)

3단계 허용식품에 아래 추가

닭고기, 소고기, 돼지고기 등 육류(지방이 적은 부위로), 토마토, 플레인요거트 먹을 때 블루베리 등 베리류 허용, 고강도 운동이나 근력운동을 한 날은 운동 전후 고구마 1개 또는 바나나 1개 허용

약 60개의 질문!

YES

NO

다이어트할 때 필수 시청
'이 음식 먹어도 될까요' 최종판

← 오픈채팅 검색

Q 스위치온

그룹 채팅 1:1 채팅

추천



보이스룸 진행중



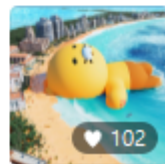
참여코드 채팅방 제외

박용우 스위치온 다이어트 함께해요

#다이어트 #박용우다이어트 #스위치온다이어트 ...



1100명 | 방금 대화



💡 탐나는 스위치온다이어트 💡

#스위치온 #다이어트 #스위치온다이어트 #다이...



186명



🔗 스위치온 다이어트 🔗

#스위치온 #다이어트 #응원 #격려



99명 | 방금 대화



스위치온 다이어트

#스위치온다이어트



다이어트 시청 '이 음식 먹으면 살 뺀다' 최종판

- 기본규칙
1. 하루에 물 8컵 이상(2리터)을 섭취한다.
 2. 수면시간은 최소 6시간 이상을 확보하고, 자정-오전 4시는 필수로 포함한다.
 3. 필요한 영양제를 꼭 섭취한다.
 4. 간헐적 단식을 실시한다.(기본 14시간 단식-10시간 식사)
 5. 저녁식사는 취침 4시간 이전에 끝낸다.
 6. 주 4회 이상 규칙적으로, 고강도 운동을 실시한다.
 7. 오래 기만이 남아있지 않는다.

1단계: 1~3일차 - 지방 대사 스위치 켜기
 1~3일차는 장에 유식을 주고, 생채리들을 개선하는 것을 목표로 한다.
 하루 4번(아침, 점심, 오후간식, 저녁) 단백질 음료만 섭취한다.(물 또는 무가당 두유와 섞
 심한 허기로 힘이 들면 녹황색 채소, 두부, 코코넛 오일, 올리브 오일, 무가당 플레인요거트
 아침공복에 유산균을 챙겨먹어야 한다.
 하루 60분 이상 걸을 수 있도록 노력한다.
 * 기존의 물 상태가 좋지 않았거나 카페인 중독이었을 경우 심한 무기력감, 두통, 현기증
 으면 대부분 사라지나, 극심한 어지럼증, 구토증세 등으로 일상생활이 어려울 정도라면 어
 기도 된다. 몸상태를 보고 결정해야 한다.

2단계: 4~7일차 - 렙틴 저항성과 지방간 개선
 허용식품에 콩류, 견과류, 약간의 김치, 오전 중 블랙커피 1잔, 우유 2잔과 짜지 않은 천연
 하루 2번(아침, 오후간식) 단백질 음료를 섭취한다.
 점심식사는 저탄수화물 식단으로, 4~7일차와 동일하게 실시한다.
 저녁식사는 무탄수화물 식단으로, 단백질이 충분한 식사로 등푸른 생
 등을 배부르게 먹는다.
 주 1회 24시간 단식을 실시한다. 자정부터 24시간 굶는 것이 아니라
 상적으로 먹는 것이다.

3단계: 2주차 - 간헐적 단식과 인슐린 저항성 개선
 허용식품에 콩류, 견과류, 약간의 김치, 오전 중 블랙커피 1잔, 우유 2잔과 짜지 않은 천연
 하루 2번(아침, 오후간식) 단백질 음료를 섭취한다.
 점심식사는 저탄수화물 식단으로, 4~7일차와 동일하게 실시한다.
 저녁식사는 무탄수화물 식단으로, 단백질이 충분한 식사로 등푸른 생선, 가금육 굶지 않고
 등을 배부르게 먹는다.
 주 1회 24시간 단식을 실시한다. 자정부터 24시간 굶는 것이 아니라 이른 저녁식사 후 익
 상적으로 먹는 것이다.

단식 하는 날은 고강도 운동은 피하되, 적절한 수준의 운동을 실시하면 효과가 더 좋다. 단
 4단계: 3주차 - 대사유연성 회복과 전진대사 최적화
 허용식품에 당호박, 토마토, 방울토마토, 밤, 바나나, 베리류 과일 등이 추가된다. 알 조절;
 하루 2번(아침, 오후간식) 단백질 음료를 섭취한다.
 점심식사는 저탄수화물 식단, 저녁식사는 무탄수화물 식단으로 2주차와 동일하게 실시한
 고강도 운동을 했다면 바나나와 고구마 등의 탄수화물 간식으로 가능하나, 추천하지 않는다
 주 2회 24시간 단식을 실시한다. 단, 이를 연속으로 수행하지 않는다.
 단식 후 고단백 식사를 충분히 해주어야 근육실이 방지된다. 3주차를 종료하고 나서 근육
 다면 4주차로 진행하고, 아니라면 다시 수행한다.

5단계: 4주차 - 업그레이드 제지방 감량 극대화하기
 허용식품에 과일이 추가된다(하루 1개로 제한) 하루 2번(아침, 오후간식) 단백질 음료로 섭취한다.
 점심식사와 저녁식사 모두 저탄수화물 식단을 한다. 주 3회 24시간 단식을 실시한다. 단, 연달아 수행하지 않는다.

- 금지 식품
1. 설탕, 액상과당
 2. 밀가루 음식

← 오픈채팅 검색

그룹 채팅 1:1 채팅

추천 ☐ 보이스룸 진행중 ☐ 참여코드 채팅방 제외

박용우 스위치온 다이어트 함께해요

#다이어트 #박용우다이어트 #스위치온다이어트 ...

102명

📍 탐나는 스위치온다이어트 📍

#스위치온 #다이어트 #스위치온다이어트 #다이...

186명

스위치온 다이어트

#스위치온 #다이어트 #응원 #격려

99명 | 발급 대화

스위치온 다이어트

#스위치온다이어트

다들의 강력한 반발(?)로 2주차부터 오전 중 블랙커피는 가능하다.
 참으나, 제로콜라 등 강한 단맛을 주는 것은 피한다. 4주간 단맛 자체를 멀리하도록 한다.

백질 음료로 대체하며, 혼용 음료는 물이나 무가당 두유 제외 다른 음료 대체 안됨.심한 허기로
 올리브 오일, 무가당 플레인요거트 등은 허용.

단백질 음료를 섭취, 점심식사는 저탄수화물 식단으로, 잡곡밥 반공기와 두부, 생선, 화, 해산물,
 등은 허용하며, 아침 저녁으로 유산균, 종합비타민, 오메가3, 비타민C, CoQ10 등의 보

김치, 오전 중 블랙커피 1잔, 우유 2잔과 짜지 않은 천연 치즈가 추가, 하루 2번(아침, 오후간
 화물 식단으로, 4~7일차와 동일하게 실시. 저녁식사는 무탄수화물 식단으로, 단백질이 충분
 이거나 삶은 고기류(수육, 사브사브 등) 등을 허기가 느껴지지 않을 정도로 섭취.

토마토, 밤, 바나나, 베리류 과일 등이 추가되며, 알 조절은 필수. 하루 2번(아침, 오후간식) 단
 식으로, 4~7일차와 동일하게 실시. 고강도 운동을 했다면 바나나
 와 고구마 등의 탄수화물 간식으로 가능하나, 추천하지 않는다

하루 2번(아침, 오후간식) 단백질 음료를 섭취. 점심식사와 저녁식사 모두 저탄수
 화물 식단으로, 4~7일차와 동일하게 실시. 고강도 운동을 했다면 바나나와 고구마 등의 탄수화물 간식도 가능하나, 추천하지 않는다

키오릭스, 양파, 마늘, 고춧가루, 식초, 후추, 강황, 허브, 양배추, 무, 당근, 오이, 브로콜리,
 1일, 아보카도오일, 냉압착 들기름, 녹차, 허브티, 무가당 플레인요거트, 두부, 연두부, 무가

해조류(미역, 다시마, 톳), 버섯류, 와사비, 저염간장(약간), 두부발장, 김치(약간), 달걀,
 1개, 오징어, 낙지, 문어), 닭고기(껍질 벗긴 속살), 삶은 돼지고기 살코기(수육), 소고기 살

S, 퀴노아, 콩류(검은콩, 병아리콩, 원두콩, 렌틸콩 등), 견과류(한 줌), 블랙커피(오전 중 한

닭고기, 소고기, 돼지고기 등 육류(지방이 적은 부위로), 토마토, 플레인요거트 먹을 때 블루베리 등 베리류 허용, 고강도 운동이나 근
 력운동을 한 날은 운동 전후 고구마 1개 또는 바나나 1개 허용

박용우봇의 필요성

박용우봇

스위치온 관련된 것은
무엇이든 물어보세요!

트러플 오일도 되네



스위치온 다이어트의 식품 허용유무를 알려주고 보유한 재료를 기반으로 만들 수 있는 다이어트 레시피 추천



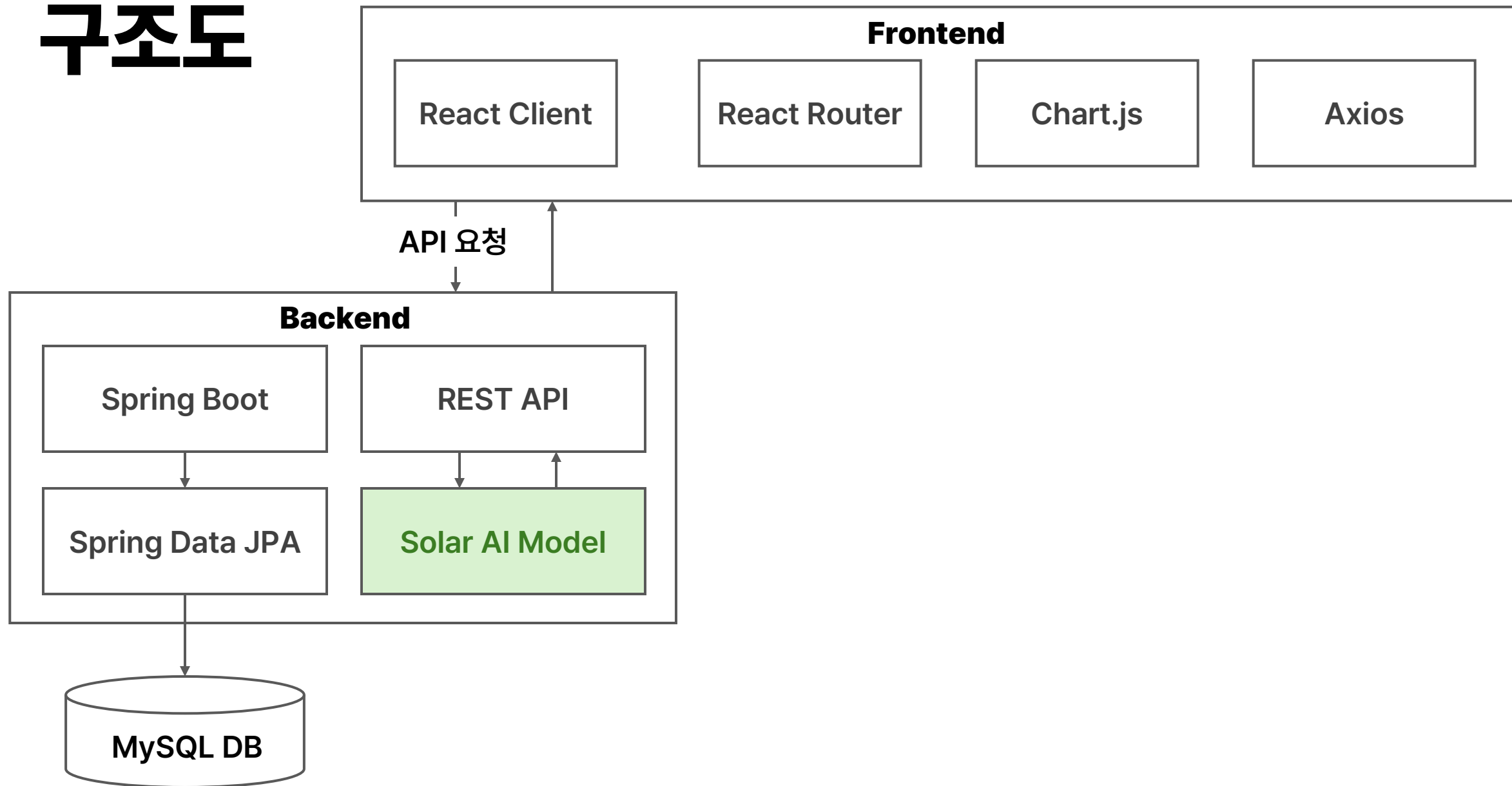
식사시간 알림 ☐

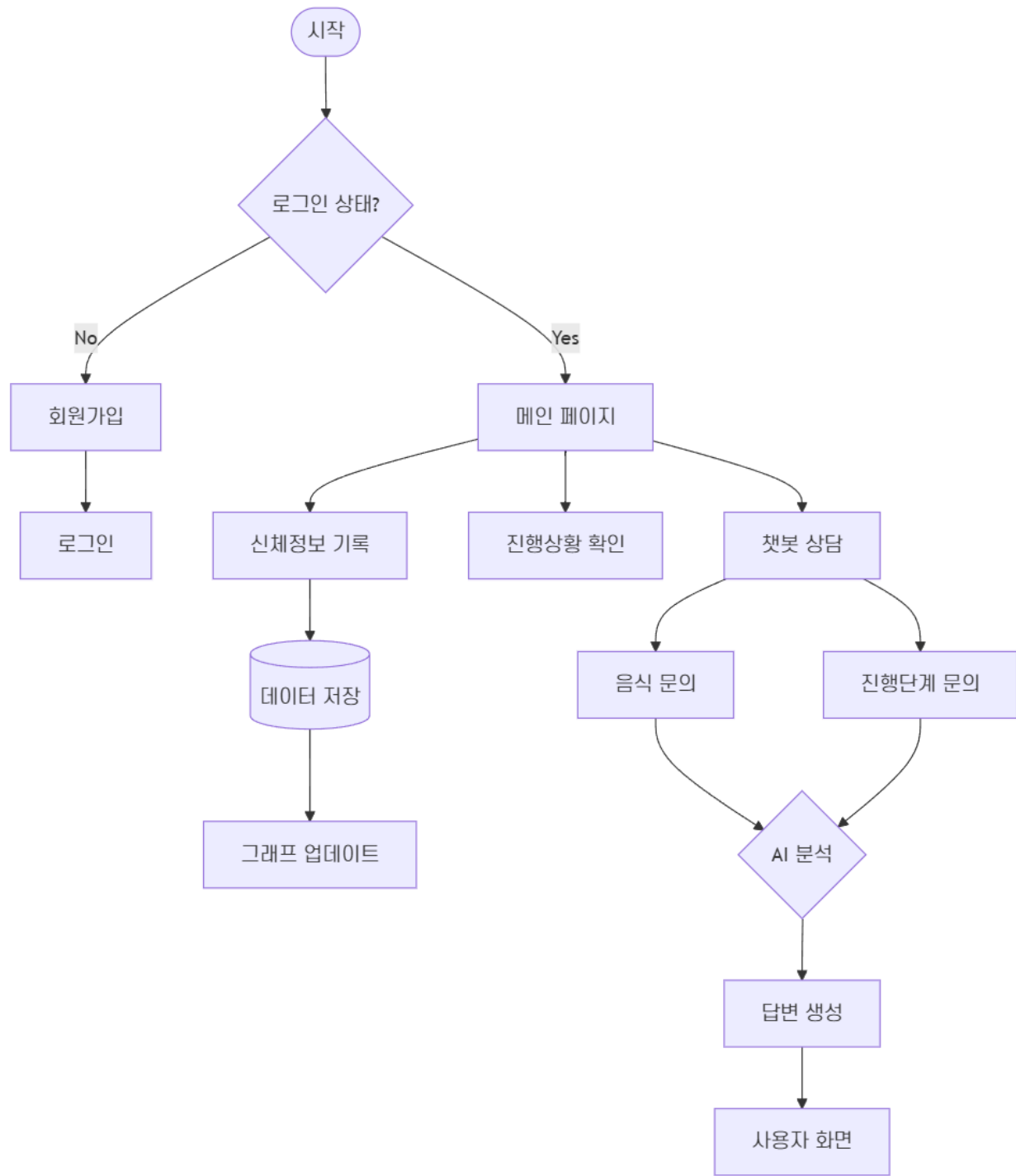
챗봇을 통한 허용 식품 확인 및 식단 가이드 ☒

(추가)영양성분표 OCR 판독 ☐

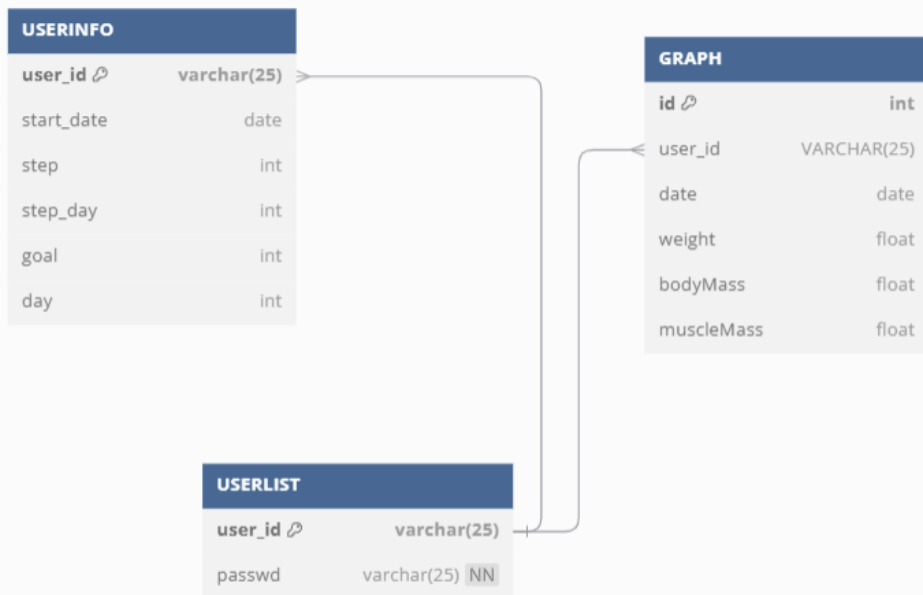
(추가)개인 맞춤형 단백질 및
탄수화물 섭취량 계산 ☐

구조도





ERD



userlist

사용자 리스트 저장을 위한 테이블

- user_id(pk): varchar(25)
- password: varchar(25)

userInfo

사용자 개별 정보를 저장하기 위한 테이블

- user_id(pk, fk): varchar(25)
- start_date: date
- step(다이어트 단계): int
- step_day(단계의 며칠차): int
- goal(목표 체중): float
- day: int

graph

사용자 체중/근육량/지방량 변화 저장을 위한 테이블

- id(pk): int
- user_id(fk): varchar
- date(기록을 저장하는 날짜): date
- weight: float
- bodyMass: float
- muscleMass: float

JPA- 연관관계

@Entity 17개 사용 위치 dlwlsghmlgh *

@Table(name = "userlist")

public class Userlist {

@Id

@Column(name = "user_id", nullable = false, length = 25)

private String userId;

@Column(name = "passwd", nullable = false, length = 25) 2개 사용 위치

private String passwd;

@Entity 20개 사용 위치 dlwlsghmlgh +1

@Table(name = "userinfo")

public class Userinfo {

@Id

@Column(name = "user_id", nullable = false, length = 25)

private String userId;

@Column(name = "start_date") 3개 사용 위치

private LocalDate startDate;

@Column(name = "step") 9개 사용 위치

private Integer step;

@Column(name = "day") 3개 사용 위치

private Integer day;

@MapsId 2개 사용 위치

@OneToOne(fetch = FetchType.LAZY, optional = false)

@JsonIgnore

@JoinColumn(name = "user_id", nullable = false)

private Userlist userList;

@Column(name = "goal") 2개 사용 위치

private Float goal;

@Column(name = "step_day") 8개 사용 위치

private Integer stepDay;

1. userinfo의 user_id는 userlist의 user_id를 참조
→ 1:1 관계 (@OneToOne)

2. graph의 user_id는 userlist의 user_id를 참조
→ n:1관계 (@ManyToOne)

@Entity 21개 사용 위치 dlwlsghmlgh +1

@Table(name = "graph")

public class Graph {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id", nullable = false)

private Integer id;

@Column(name = "date") 2개 사용 위치

private LocalDate date;

@Column(name = "weight") 2개 사용 위치

private Float weight;

@Column(name = "body_mass") 2개 사용 위치

private Float bodyMass;

@Column(name = "muscle_mass") 2개 사용 위치

private Float muscleMass;

@ManyToOne(fetch = FetchType.LAZY) 2개 사용 위치

@JoinColumn(name = "user_id", referencedColumnName = "user_id", nullable = false)

@JsonBackReference // 반대쪽 참조 필드로 직렬화 제외

private Userlist user;

프레임워크 / 데이터 관리 / 빌드도구

Spring Boot / MySQL, Spring Data JPA / Gradle

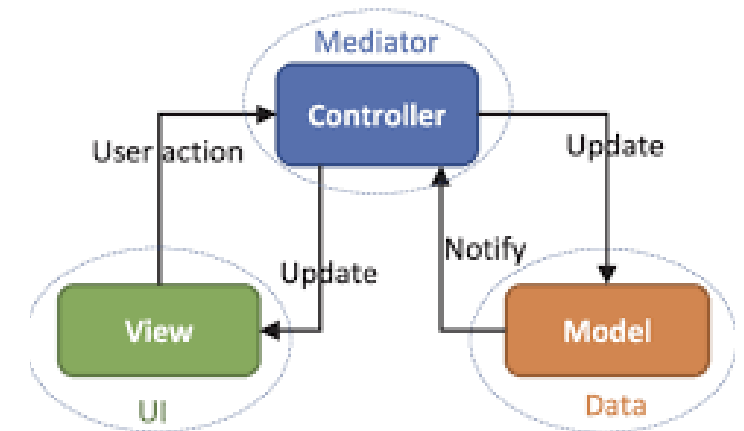


아키텍처 패턴

MVC: 컨트롤러, 서비스, 레포지토리 계층으로 나누어 모듈화된 구조로 구성

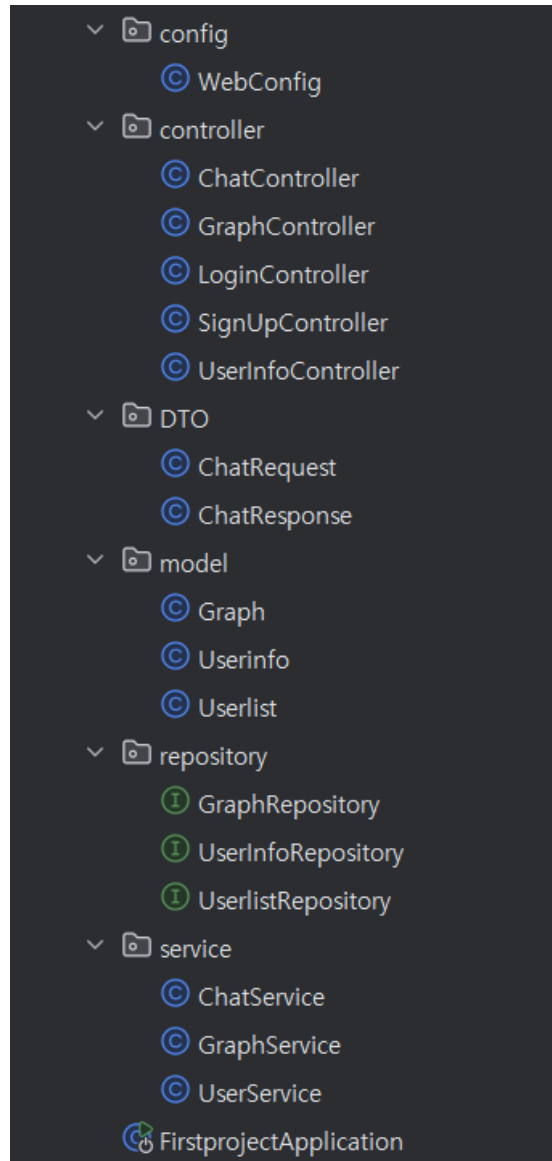
API 통신

RESTful API: 프론트와의 데이터 교환, AI 서버와 데이터 교환



Model-View-Controller(MVC) design pattern.

<MVC 구조>



Config

: 전역적인 설정을 관리하는 클래스(프론트, AI와 통신 허용)

Controller

ChatController

: 프론트 요청을 받아 AI서버와 통신

GraphController

: 사용자 그래프 데이터 관리

LoginController / SignUpController

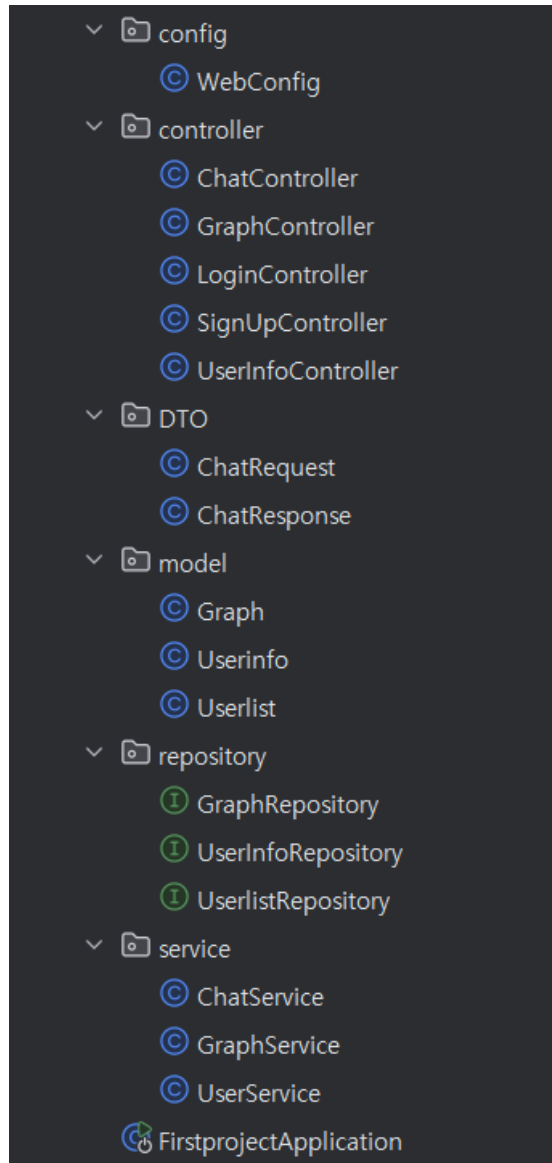
: 사용자 로그인/회원가입 요청 처리

UserInfoController

: 사용자 마이페이지 데이터 조회

DTO

: 챗봇에서 요청데이터, 응답데이터 저장하는 클래스



Model / Repository

Graph

: 사용자 그래프 데이터를 나타냄

Userinfo

: 사용자 추가 정보/데이터 관리

Userlist

: 사용자 계정 정보 관리

Service

ChatService

: WebClient 사용해 AI 서버와 통신

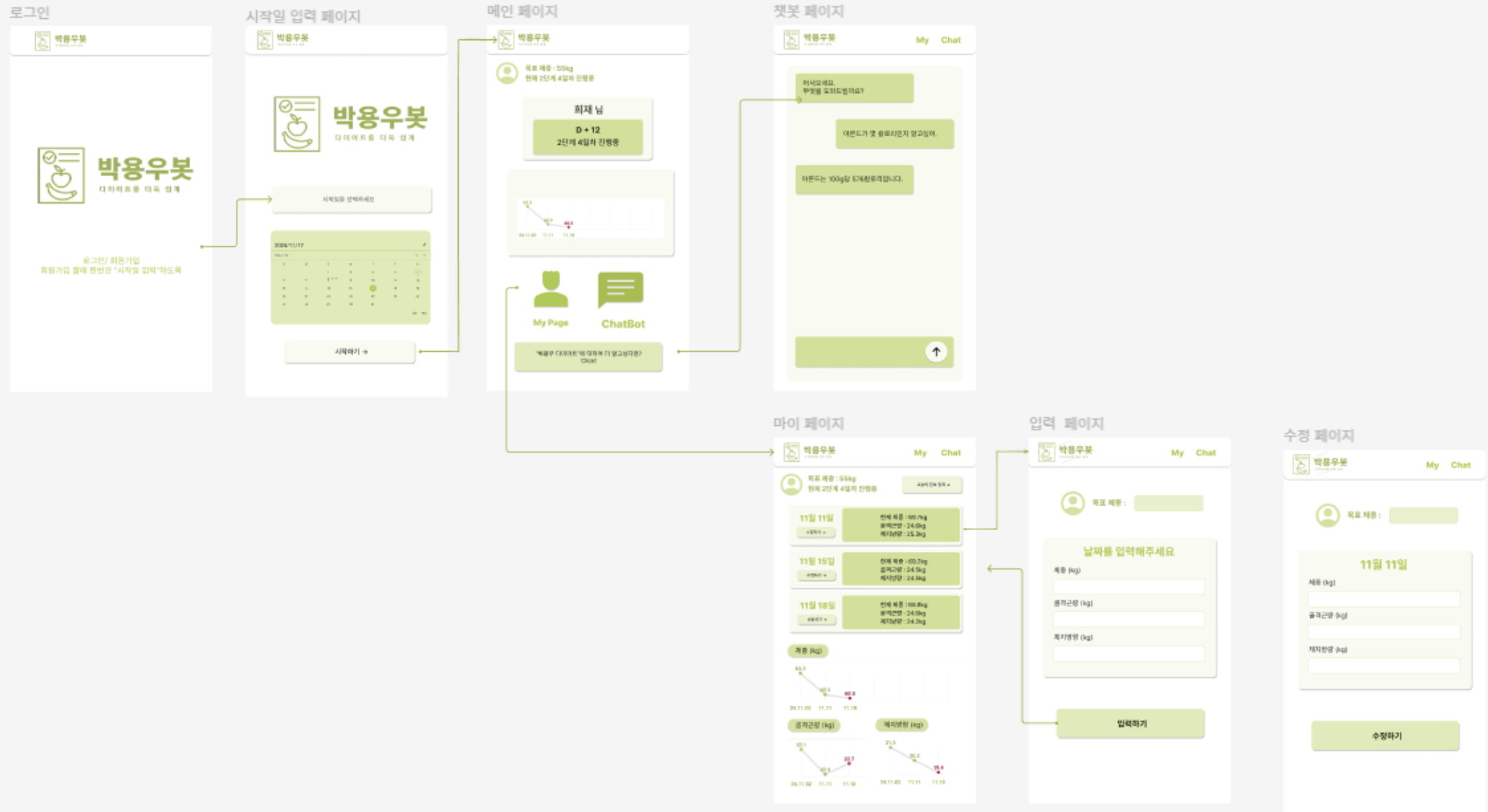
GraphService

: 그래프기록 추가, 조회, 수정 관리

UserService

: 로그인, 회원가입, 정보조회 관리

FIGMA



```

import React, { useState, useEffect, useContext } from "react";
import { Link } from "react-router-dom";
import { Line } from "react-chartjs-2";
import axios from "axios";
import { DataContext } from "../App";
import {
  Chart as ChartJS,
  CategoryScale,
  LinearScale,
  PointElement,
  LineElement,
  Title,
  Tooltip,
  Legend,
} from "chart.js";

ChartJS.register(CategoryScale, LinearScale, PointElement, LineElement, Title, Tooltip, Legend);

```

필요한 라이브러리 및 모듈 가져오기

```

const MainPage = () => {
  const [goalWeight, setGoalWeight] = useState(0);
  const [currentStep, setCurrentStep] = useState("");
  const [daysInPhase, setDaysInPhase] = useState(0);
  const [weightDataLocal, setWeightDataLocal] = useState([]);
  const { setWeightData } = useContext(DataContext);
  const userId = "user001";

  useEffect(() => {
    const fetchUserData = async () => {
      try {
        const userResponse = await axios.get(`/api/user/mypage/user-record/${userId}`);
        const userData = userResponse.data;
        setGoalWeight(userData.goal);
        setCurrentStep(`${userData.step}단계`);
        setDaysInPhase(userData.day);

        const graphResponse = await axios.get(`/api/user/mypage/graph-records/${userId}`);
        const graphData = (graphResponse.data || []).map((record) => ({
          date: record.date || "N/A",
          weight: record.weight || 0,
          muscleMass: record.muscleMass || 0,
          bodyFat: record.bodyMass || 0,
        }));
        console.log("Graph Records Data:", graphData);

        setWeightDataLocal(graphData);
        setWeightData(graphData);
      } catch (error) {
        console.error("Error fetching data:", error);
      }
    };
    fetchUserData();
  }, [userId, setWeightData]);

  const weightChartData = {
    labels: weightDataLocal.map((entry) => entry.date),
    datasets: [
      {
        label: "체중 (kg)",
        data: weightDataLocal.map((entry) => entry.weight),
        borderColor: "rgb(163, 201, 120)",
        backgroundColor: "rgba(75, 192, 192, 0.2)",
        borderWidth: 2,
      },
    ],
  };
};

```

api 호출

```

</p>
<p className="text-gray-500">
  {currentStep} {daysInPhase}일 도전 중!
</p>
</section>

<section
  className="w-full max-w-md shadow-lg rounded-lg p-6 flex items-center justify-center"
  style={{ backgroundColor: "rgb(249,251,242)" }}
>
  <div className="w-full">
    <h3 className="text-center text-lg font-bold mb-4 text-gray-700">체중 현황</h3>
    <img data={weightChartData} />
  </div>
</section>

<section className="w-full max-w-md flex justify-around">
  <link to="/mypage">
    <button
      className="px-6 py-2 text-black rounded-lg shadow-md hover:bg-green-600"
      style={{ backgroundColor: "rgb(221,235,200)" }}
    >
      My Page
    </button>
  </link>
  <link to="/chatbot">
    <button
      className="px-6 py-2 text-black rounded-lg shadow-md hover:bg-green-600"
      style={{ backgroundColor: "rgb(221,235,200)" }}
    >
      ChatBot
    </button>
  </link>
</section>

<section
  className="w-full max-w-md shadow-lg rounded-lg p-4 text-center"
  style={{ backgroundColor: "rgb(221,235,200)" }}
>
  <p className="text-gray-700">
    "박용우 다이어트"에 대해 더 알고 싶다면? (" ")
    <a
      href="https://namu.wiki/w/%EC%A4%BC%9C%B9%98%EC%98%A8%20%EB%A4%EC%98%A4%EC%96%B4%ED%8A%B8"
      className="text-blue-500 underline hover:text-blue-700"
      target="_blank"
      rel="noopener noreferrer"
    >
      Click!
    </a>
  </p>
</section>
</div>
);

```

목표 체중 63kg를 향해

9일차 3단계 도전 중!

체중 변화



My Page

ChatBot

"박용우 다이어트"에 대해 더 알고 싶다면? [Click!](#)

피그마 디자인에 맞게 ui를 구성

- 사용자 목표체중 및 현재 진행 상황
- 체중 변화 그래프
- 페이지 이동 버튼
- 추가 정보 제공

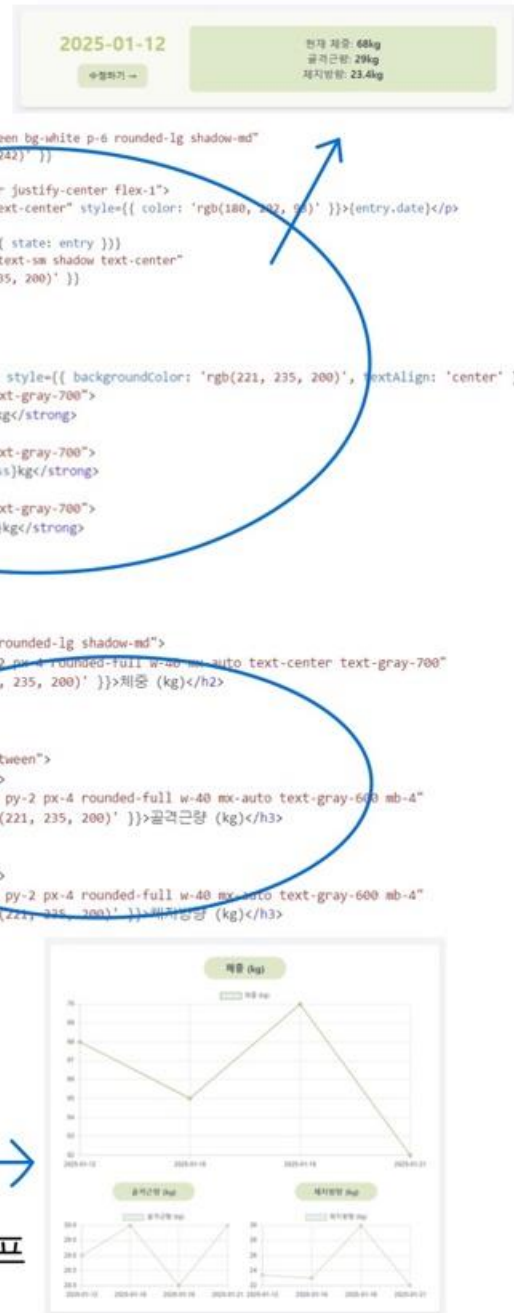
필요한 라이브러리 및 모듈 가져오기

api 호출

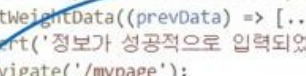
체중, 골격근량, 체지방량에
따른 그래프 형성

ui 구성

- 날짜별 진행 상황
- 체중, 골격근량, 체지방량 변화 그래프




```
const response = await axios.post(`/api/user/set/record`, payload);
setWeightData((prevData) => [...prevData, { ...response.data }]);
alert('정보가 성공적으로 입력되었습니다.');
```



```

navigate('/mypage');
catch (error) {
  console.error('Error submitting weight data:', error);
  alert('정보 입력 중 오류가 발생했습니다.');
```

api를 통해 백

```

}
finally {
  setIsLoading(false);
}
};
```

api를 통해 백엔드로 입력된 정보 전송

날파를 선택해주세요
2025. 1. 21.

체중 (kg)
체중 입력

골격근량 (kg)
골격근량 입력

체지방량 (kg)
체지방량 입력

입력하기

- 날짜입력
- 몸무게, 골격근량, 체지방량 입력
- 입력하기 버튼

```

import React, { useState, useEffect } from 'react';
import { useLocation, useNavigate } from 'react-router-dom';
import axios from 'axios';
import DatePicker from 'react-datepicker';
import 'react-datepicker/dist/react-datepicker.css';

const EditPage = () => {
  const { state: initialData } = useLocation();
  const navigate = useNavigate();
  const [formData, setFormData] = useState({
    date: initialData?.date ? new Date(initialData.date) : new Date(),
    weight: initialData?.weight || '',
    muscle: initialData?.muscleMass || '',
    fat: initialData?.bodyMass || '',
  });
  const [isLoading, setIsLoading] = useState(false);
  const recordId = initialData?.id;

  const handleChange = (name, value) => {
    if (value === "" || value === null) {
      setFormData({ ...formData, [name]: "" });
      return;
    }

    if (value < 0) {
      alert("값은 음수일 수 없습니다.");
      return;
    }

    setFormData({ ...formData, [name]: value });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    setIsLoading(true);
    try {
      const payload = {
        id: recordId,
        date: formData.date.toISOString().split('T')[0],
        weight: formData.weight || 0,
        bodyMass: formData.fat || 0,
        muscleMass: formData.muscle || 0,
      };

      const response = await axios.put(`/api/user/modify/record/${recordId}`, payload);

      alert("수정이 성공적으로 완료되었습니다.");
      navigate("/mypage");
    } catch (error) {
      console.error("Error updating record:", error);
      alert("정보 수정 중 오류가 발생했습니다.");
    } finally {
      setIsLoading(false);
    }
  };
};

```

```

return (
  <div className="min-h-screen bg-gray-100 p-6">
    <section className="space-y-8">
      <div
        className="space-y-4 bg-white p-6 rounded-lg shadow-md"
        style={{ backgroundColor: "rgb(249, 251, 242)" }}
      >
        <div className="flex flex-col items-center mb-4">
          <p className="text-center text-gray-700 mb-2">{
            formData.date.toISOString().split('T')[0]}</p>
          <DatePicker
            selected={formData.date}
            onChange={(date) => handleChange("date", date)}
            dateFormat="yyyy-MM-dd"
            inline
            className="rounded-lg border border-gray-300 shadow-sm"
          />
        </div>
        <div>
          <label className="block text-lg font-medium text-gray-700">체중 (kg)</label>
          <input
            type="number"
            name="weight"
            value={formData.weight}
            onChange={(e) => handleChange("weight", e.target.value)}
            placeholder={`현재 체중: ${initialData?.weight || ""}`}
            className="w-full mt-1 px-4 py-2 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-2 focus:ring-green-500"
          />
        </div>
        <div>
          <label className="block text-lg font-medium text-gray-700">골격근량 (kg)</label>
          <input
            type="number"
            name="muscle"
            value={formData.muscle}
            onChange={(e) => handleChange("muscle", e.target.value)}
            placeholder={`현재 골격근량: ${initialData?.muscleMass || ""}`}
            className="w-full mt-1 px-4 py-2 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-2 focus:ring-green-500"
          />
        </div>
      </div>
      <div>
        <label className="block text-lg font-medium text-gray-700">체지방량 (kg)</label>
        <input
          type="number"
          name="fat"
          value={formData.fat}
          onChange={(e) => handleChange("fat", e.target.value)}
          placeholder={`현재 체지방량: ${initialData?.bodyMass || ""}`}
          className="w-full mt-1 px-4 py-2 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-2 focus:ring-green-500"
        />
      </div>
      <div>
        <form onSubmit={handleSubmit} className="flex justify-center mt-6">
          <button
            type="submit"
            className="w-1/2 px-4 py-2 font-medium rounded-full shadow-md hover:bg-green-300"
            style={{ backgroundColor: "rgb(221, 235, 200)" }}
            disabled={isLoading}
          >
            {isLoading ? "수정 중..." : "수정하기"}
          </button>
        </form>
      </div>
    </section>
  </div>
);

```

2025. 1. 12.

January 2025						
Su	Mo	Tu	We	Th	Fr	Sa
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

체중 (kg)
68

골격근량 (kg)
29

체지방량 (kg)
23.4

수정하기

api를 통해 백엔드로 수정된 정보 전송

ui 구성

- 날짜 수정
- 몸무게, 골격근량, 체지방량 수정
- 수정하기 버튼


```

1 # 필요한 임포트
2 import openai
3 from fastapi import FastAPI, HTTPException, status, Response
4 from fastapi.middleware.cors import CORSMiddleware
5 from fastapi.responses import HTMLResponse
6 from pydantic import BaseModel, Field
7 from typing import List, Dict, Optional
8 from datetime import datetime
9 import uvicorn
10 import uuid
11 from pyngrok import ngrok
12 import nest_asyncio
13
14 # API 설정
15 openai.api_key = "up_JB2rvyv0IYlmdY94c0h8z9h9JzwfP"
16 openai.api_base = "https://api.upstage.ai/v1/solar"
17
18 # Pydantic 모델
19 class ChatRequest(BaseModel):
20     content: str = Field(..., description="사용자 메시지")
21     user_id: str = Field(..., description="사용자 ID")
22     start_date: datetime = Field(..., description="시작 날짜")
23     step: int = Field(..., description="현재 단계", ge=1, le=5)
24     day: int = Field(..., description="현재 일자", ge=1)
25
26 class Config:
27     json_schema_extra = {
28         "example": {
29             "content": "오늘 식단 추천해주세요",
30             "user_id": "user123",
31             "start_date": "2025-01-01",
32             "step": 1,
33             "day": 1
34         }
35     }
36
37 class ChatResponse(BaseModel):
38     success: bool
39     message: str
40     data: Optional[dict]
41     timestamp: datetime
42
43 class DietPhase:
44     def __init__(self, phase_num, days, title, goal, guidelines, allowed_foods, restrictions, fasting_rule=None):
45         self.phase_num = phase_num
46         self.days = days
47         self.title = title
48         self.goal = goal
49         self.guidelines = guidelines
50         self.allowed_foods = allowed_foods
51         self.restrictions = restrictions
52         self.fasting_rule = fasting_rule
53
54 class DietKnowledgeBase:
55     def __init__(self):
56         self.phases = self.initialize_phases()
57         self.general_rules = {

```

Solar API 불러오기

Q, A형식이 아닌 단계별 형식으로 정리하여
지식베이스 (domain knowledge) 구축

```

58         self.general_rules = {
59             "supplements": ["유산균", "종합비타민", "오메가3", "비타민C", "CoQ10"],
60             "exercise": "하루 60분 이상 걷기",
61             "prohibited_foods": [
62                 "설탕류(정백당, 액상과당)", "청량음료", "커피믹스", "과자", "도넛",
63                 "아이스크림", "주스", "당분이 첨가된 우유/두유",
64                 "밀가루 음식(빵, 케이크, 파스타, 라면)", "술",
65                 "포화지방이 많은 음식(삼겹살, 갈비, 곱창)", "트랜스지방이 많은 음식",
66                 "짠 음식(소금, 양념장, 젓갈류)"
67             ]
68         }
69
70     def initialize_phases(self):
71         return [
72             1: DietPhase(
73                 phase_num=1,
74                 days="1-3일자",
75                 title="지방 대사 스위치 켜기",
76                 goal="장 휴식과 생체리듬 개선",
77                 guidelines=[
78                     "하루 4번(아침, 점심, 오후간식, 저녁) 단백질 음료 섭취",
79                     "아침 공복에 유산균 섭취",
80                     "하루 60분 이상 걷기",
81                     "물 또는 무가당 두유만 허용",
82                     "카페인 섭취 금지"
83                 ],
84                 allowed_foods=[
85                     "단백질 셰이크", "녹황색 채소", "두부", "코코넛 오일",
86                     "올리브 오일", "무가당 플레인요거트", "녹차", "허브티",
87                     "양파", "마늘", "고춧가루", "식초", "후추", "강황", "허브",
88                     "양배추", "무", "당근", "오이", "브로콜리", "파프리카",
89                     "아보카도", "연두부", "무가당 두유"
90                 ],
91                 restrictions=[
92                     "물 또는 무가당 두유 외 다른 음료 제한",
93                     "카페인 섭취 금지",
94                     "커피 섭취 금지",
95                     "심한 허기 시에만 허용 식품 섭취 가능"
96                 ]
97             ),
98             2: DietPhase(
99                 phase_num=2,
100                 days="4-7일자",
101                 title="렙틴 저항성과 지방간 개선",
102                 goal="렙틴 저항성 개선",
103                 guidelines=[
104                     "하루 3번(아침, 오후간식, 저녁) 단백질 음료 섭취",
105                     "점심은 저탄수화물 식단",
106                     "영양제 섭취(아침, 저녁)",
107                     "하루 60분 이상 걷기 유지",
108                     "카페인 섭취 금지",
109                     "커피 섭취 금지"

```

```

def get_system_prompt(self) -> str:
    prompt_template = ""당신은 박용우 다이어트 전문가입니다. 다음의 정보를 기반으로 상담을 진행합니다:

```

다이어트 단계별 정보:

{phase_info}

일반 규칙:

- 보충제: {supplements}
- 운동: {exercise}
- 금지 식품: {prohibited_foods}

주의사항:

1. 단계별 특성을 고려하여 맞춤형 조언을 제공해주세요.
2. 허용 식품과 제한 식품을 명확하게 구분하여 설명해주세요.
3. 답변할 때는 항상 근거를 함께 설명해주세요.
4. 단계별 목표와 가이드라인을 참고하여 구체적인 실천 방안을 제시해주세요.
5. 답변은 간결하고 정확하게 하되, 친절하게 해주세요.""

```

phase_info = ""
for phase_num in range(1, 6):
    phase = self.get_phase_info(phase_num)
    if phase:
        phase_info += f"Wn{phase['days']} - {phase['title']}Wn"
        phase_info += f"목표: {phase['goal']}Wn"
        phase_info += f"가이드라인:Wn" + "Wn".join(f"- {g}" for g in phase['guidelines'])
        if phase['fasting_rule']:
            phase_info += f"Wn단식 규칙: {phase['fasting_rule']}"
        phase_info += "Wn"

return prompt_template.format(
    phase_info=phase_info,
    supplements=", ".join(self.general_rules["supplements"]),
    exercise=self.general_rules["exercise"],
    prohibited_foods="Wn- ".join(self.general_rules["prohibited_foods"])
)

```

class DietChatbot:

```

def __init__(self, knowledge_base: DietKnowledgeBase):
    self.knowledge_base = knowledge_base
    self.messages = [
        {
            "role": "system",
            "content": knowledge_base.get_system_prompt()
        }
    ]

```

def send_message(self, request: ChatRequest) -> str:

```

try:
    # 현재 단계 정보 가져오기
    current_phase = self.knowledge_base.get_phase_info(request.step)

    # 컨텍스트 생성
    context = (
        f"다음 정보를 바탕으로 사용자의 질문에 답변해주세요:WnWn"
        f"[현재 단계 정보]Wn"
        f"- 단계: {request.step}단계 ({current_phase['title']})Wn"

```

프롬프트 엔지니어링

메인 페이지



박용우봇

다 이 어 트 를 더 욱 쉽 게

[Log In](#)[Sign In](#)

회원가입

회원가입

아이디

비밀번호

비밀번호 확인

목표 체중 (kg)

시작일을 선택하세요

회원가입하기 →

로그인 페이지

로그인

아이디

비밀번호

마이 페이지

목표 체중 63kg를 향해

9일차 3단계 도전 중!

체중 변화



My Page

ChatBot

"박용우 다이어트"에 대해 더 알고 싶다면? [Click!](#)

상세 페이지



목표 체중: 63kg
현재 단계: 3단계
진행 중인 일수: 9일

[오늘의 정보 입력 →](#)

2025-01-12

[수정하기 →](#)

현재 체중: 68kg
골격근량: 29kg
체지방량: 23.4kg

2025-01-16

[수정하기 →](#)

현재 체중: 65kg
골격근량: 30kg
체지방량: 23kg

2025-01-18

[수정하기 →](#)

현재 체중: 70kg
골격근량: 28kg
체지방량: 30kg

2025-01-21

[수정하기 →](#)

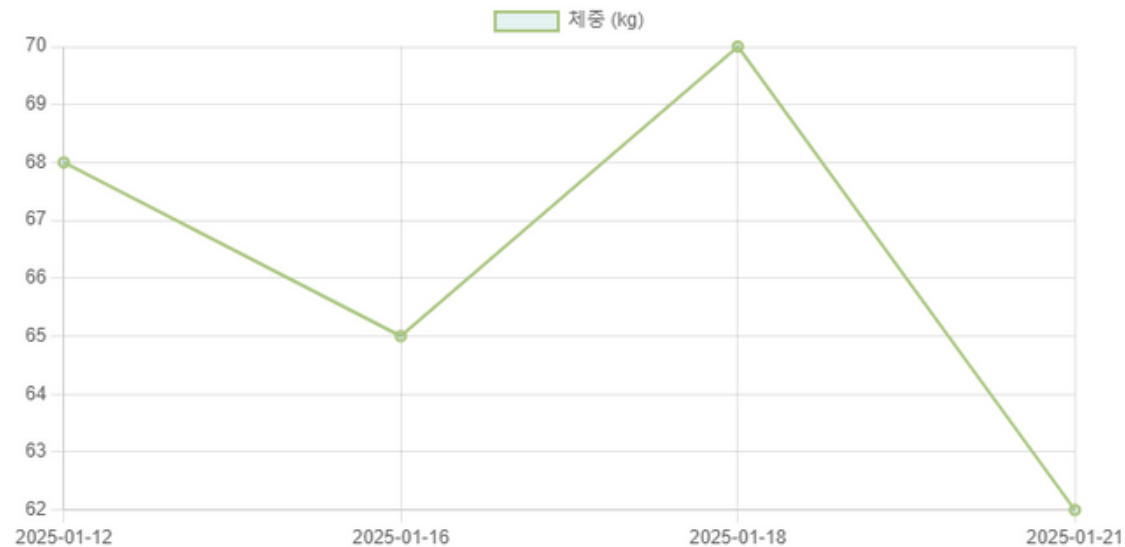
현재 체중: 62kg
골격근량: 30kg
체지방량: 22kg

수정하기 →

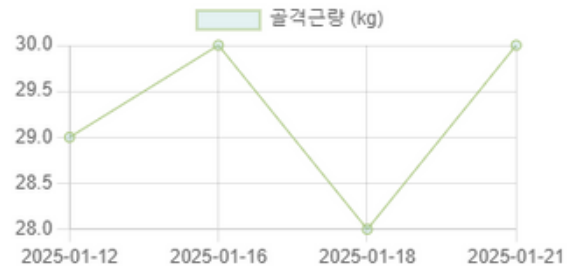
체지방량: 22kg

체중 변화 그래프

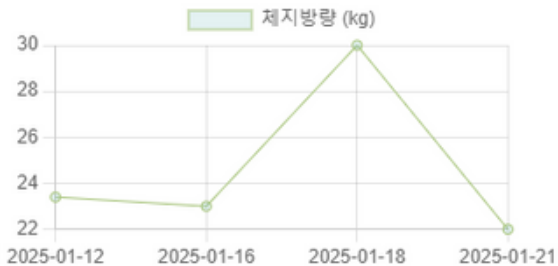
체중 (kg)



골격근량 (kg)



체지방량 (kg)



체중 기록

2025. 1. 12.

< January 2025 >						
Su	Mo	Tu	We	Th	Fr	Sa
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

체중 (kg)

68

골격근량 (kg)

29

체지방량 (kg)

23.4

수정하기

박용우봇

json 형태 output을
입력받아 출력

어서오세요. 무엇을 도와드릴까요?

현재 단계에서 아메리카노 마셔도 되나요?

네, 현재 단계에서 아메리카노는 마실 수 있습니다. 아메리카노는 블랙커피에 물을 섞은 것으로, 설탕이나 시럽 등의 첨가물이 들어가지 않는 한 칼로리가 거의 없습니다. 또한, 아메리카노에 들어간 카페인 성분은 신진대사를 촉진시켜 체지방 분해에 도움을 줄 수 있습니다. 하지만, 과도한 카페인 섭취는 불면증, 심장 두근거림, 위장 장애 등의 부작용을 일으킬 수 있으므로, 하루에 1잔 이내로 제한하는 것이 좋습니다.

라떼도 가능한가요?

죄송하지만, 라떼는 현재 단계에서 권장되지 않는 음료입니다. 라떼는 우유와 에스프레소로 만들어지며, 우유에는 탄수화물과 지방이 포함되어 있습니다. 또한, 라떼에는 일반적으로 설탕이 첨가되므로 다이어트 중에는 섭취를 제한하는 것이 좋습니다. 대신, 블랙커피나 아메리카노를 선택하여 저탄수화물 식단을 유지하는 것이 좋습니다.

짜파게티에 트러플 오일 넣어서 먹어도 되나요?

짜파게티는 탄수화물 함량이 높은 면으로 이루어져 있어 현재 단계에서는 권장되지 않는 음식입니다. 트러플 오일은 지방 함량이 높아 다이어트 중에는 섭취를 제한해야 합니다. 따라서, 짜파게티에 트러플 오일을 넣어서 먹는 것은 권장되지 않습니다. 대신, 허용된 저탄수화물 식품을 활용하여 건강한 식사를 즐기시기 바랍니다.

짜파게티에 아보카도 오일은요?

짜파게티는 탄수화물 함량이 높은 면이므로 현재 단계에서는 권장되지 않는 음식입니다. 아보카도 오일은 건강한 지방이지만, 탄수화물이 많은 짜파게티와 함께 섭취하면 탄수화물 섭취를 제한하는 모전을 상식할 수 있습니다. 대신, 허용된 저탄수화물 식품을 활용하여 건강한 식사를 즐기시기 바랍니다.

메시지를 입력하세요...

전송

Solar API 사용
MySQL의 데이터와
채팅이 json 형태로 전송

박용우봇

죄송하지만, 라떼는 현재 단계에서 권장되지 않는 음료입니다. 라떼는 우유와 에스프레소로 만들어지며, 우유에는 탄수화물과 지방이 포함되어 있습니다. 또한, 라떼에는 일반적으로 설탕이 첨가되므로 다이어트 중에는 섭취를 제한하는 것이 좋습니다. 대신, 블랙커피나 아메리카노를 선택하여 저탄수화물 식단을 유지하는 것이 좋습니다.

짜파게티에 트러플 오일 넣어서 먹어도 되나요?

짜파게티는 탄수화물 함량이 높은 면으로 이루어져 있어 현재 단계에서는 권장되지 않는 음식입니다. 트러플 오일은 지방 함량이 높아 다이어트 중에는 섭취를 제한해야 합니다. 따라서, 짜파게티에 트러플 오일을 넣어서 먹는 것은 권장되지 않습니다. 대신, 허용된 저탄수화물 식품을 활용하여 건강한 식사를 즐기시기 바랍니다.

짜파게티에 아보카도 오일은요?

짜파게티는 탄수화물 함량이 높은 면이므로 현재 단계에서는 권장되지 않는 음식입니다. 아보카도 오일은 건강한 지방이지만, 탄수화물이 많은 짜파게티와 함께 섭취하면 탄수화물 섭취를 제한하는 목적을 상실할 수 있습니다. 대신, 허용된 저탄수화물 식품을 활용하여 건강한 식사를 즐기시는 것이 좋습니다. 아보카도 오일은 샐러드 드레싱이나 요리에 활용하여 건강한 지방을 섭취할 수 있습니다.

소고기는요?

소고기는 단백질이 풍부하고 지방 함량이 낮은 육류로, 현재 단계에서 섭취할 수 있는 좋은 선택입니다. 그러나 소고기를 섭취할 때는 양념을 주의해야 합니다. 양념에 설탕이나 탄수화물이 들어간 경우가 많으므로, 소금이나 후추, 레몬즙 등 간단한 양념으로 조리하는 것이 좋습니다. 또한, 소고기를 섭취할 때는 탄수화물 섭취를 제한하기 위해 무, 양파, 상추 등과 함께 섭취하는 것이 좋습니다.

감사합니다.

시연 영상



시연 영상

어서오세요. 무엇을 도와드릴까요?



박용우봇

다 이 어 트 를 더 욱 쉽 게

Log In

Sign In