

이름: 김태훈 (Theodore Kim)

제출일: 2024-10-24

일단 읽기 전에 제가 한국어가 많이 미숙해서 최대한 열심히 써보았습니다. >_<

자동화 식사 추천 시스템

프로젝트 설명

- 사용자가 입력한 재료와 개인 정보(키, 체중)를 바탕으로 요리를 추천하는 프로그램입니다.
- 냉장고에 있는 재료를 기반으로 다양한 레시피를 제안하며, 개인의 체중 및 신체 정보를 고려하여 맞춤형 식단을 제공합니다.
- 각 재료에 대한 칼로리를 포함하여 예상 음식 칼로리를 계산합니다.

프로젝트 요구 사항

- 사용자가 보유한 재료 입력 기능
- 키와 체중 입력 기능
- 평균 체중과 비교하여 식단 조절 권장 기능
- 기본 재료의 칼로리 정보 포함
- 예상 음식 칼로리 계산 기능
- 재료에 맞는 레시피 검색 기능
- 레시피 추천 및 요리 방법 제공

기능 세부 구현

1. 재료 입력: 사용자가 현재 냉장고에 있는 재료를 입력합니다.
2. 개인 정보 입력: 사용자가 자신의 키와 체중을 입력합니다.
3. 평균 체중 비교:
 - BMI(체질량지수)를 계산하여 개인의 체중 상태를 평가합니다.
 - 평균 BMI에 비해 차이가 많이 나면, 체중 조절을 위한 권장 식단을 제안합니다.

BMI 계산

- $BMI = \text{체중(kg)} / (\text{키(m)} \times \text{키(m)})$
- BMI 기준에 따라:
 - 저체중: $BMI < 18.5$
 - 정상: $18.5 \leq BMI < 24.9$
 - 과체중: $25 \leq BMI < 29.9$
 - 비만: $BMI \geq 30$
 -
- 4. 기본 재료의 칼로리 정보: 프로그램에 기본 재료의 칼로리 정보를 사전 정의합니다.
(예:
 - 사과: 52 kcal/100g
 - 닭가슴살: 165 kcal/100g
 - 브로콜리: 34 kcal/100g 등)

5. 예상 음식 칼로리 계산:
 - 사용자가 입력한 재료와 그 양에 따라 총 칼로리를 계산합니다.
6. 레시피 추천: 입력된 재료를 기반으로 가능한 레시피를 검색하여 추천합니다.
7. 요리 방법 출력: 추천된 레시피의 조리 방법과 소요 시간을 출력합니다.
8. 식단 조절 권장: 체중 조절이 필요한 경우, 칼로리 섭취량과 영양소 비율을 고려하여 맞춤형 레시피를 제안합니다.

역할 분배

- 김태훈(Theodore Kim)
 - 코드 틀 잡고 추가 문제 해결
- 심미지(Miji Shim)
 - 리포트 작성
- 정민서(Matthew Jung)
 - 코드 작성 도움과 다른 문제 해결

코드 설명:

이 코드는 사용자의 키와 몸무게 정보를 입력받고 사용자의 BMI를 알려준다. 그리고 미리 입력해둔 재료와 칼로리를 바탕으로 원하는 재료에 맞는 레시피를 알려준다.

먼저 `Import json`은 `javascript object notation`으로 데이터를 교환하는 형식이다. (데이터 통신)

`calories_data`는 딕셔너리 형식으로 만들어 `key` 값에는 음식들이 저장되어 있고, `value` 부분엔 그 음식의 칼로리가 저장되어 있다.

`recipes_data`는 재료를 보고 레시피를 추천해 줄 수 있게끔 레시피 목록을 적어놓은 것이다 (재료나 레시피는 원하면 더 추가 가능!)

`def user_info` 함수는 먼저 키와 체중을 입력받는다 키는 `Meter`로 저장시키기 위해 `100`으로 나누고, 무게는 그대로 쓰인다. 숫자가 아닌 다른 것을 입력했을 시에는 “잘못된 입력입니다. 숫자를 입력해 주세요.”라는 문구가 뜨고 다시 돌아가서 숫자를 입력할 수 있게 된다. 유저 입력을 다 받으면 그 정보들을 바탕으로 `BMI(체지방률)`을 계산하여 바로 알려준다.

`get_ingredients_and_calories`는 처음에는 아무 정보가 없으므로 총 칼로리와 `ingredient`를 `0`으로 지정한다. `while True` 부분에선 동작하고 있을 때 `ingredient`는 재료를 엔터를 누르면 그램 수를 입력할 수 있으며 또다시 엔터를 누르면 새로운 재료 입력이 가능하게끔 뜬다. 원할 때까지 재료를 입력할 수 있으며 다 입력했을 때 “끝”이라고 입력하면 끝난다. 잘못된 양을 입력했을 시에 “잘못된 양입니다. 숫자를 입력해 주세요.”라고 뜨며 다시 입력을 할 수 있게 된다.

`return indegredients, total_calories`는 부분은 칼로리와 양을 바탕으로 값을 적은 것이다.

`def suggest_recipes`는 `recommended_recipes`라는 함수를 활용해 내가 가지고 있는 재료를 다 쓰는 레시피가 있을 경우 그 레시피를 추천해 주고 만약에 없다면 “사용할 수 있는 재료로 만들 수 있는 요리가 없습니다.”라고 뜬다.

`def save_your_info_to_json`은 사용자 정보의 데이터를 `json`이라는 정보함수에 저장을 해준다.

`def main`은 처음에 “자동화된 식사 추천 시스템에 오신 것을 환영합니다.”라고 뜨며, 몸무게와 키의 정보를 유저에게 받고 그걸 바탕으로 `BMI` 함수 계산을 프린트해준다. 그리고 총 칼로리들을 모두 불러와 몇인지를 알려주고 함수를 불러와 레시피를 불러준다.

마지막으로 `if __name__=="__main__"`는 특정 코드 블록을 프로그램의 진입점으로 설정하는 역할을 합니다. 이를 통해 해당 파일이 직접 실행될 때만 `main()` 함수를

호출하고, 다른 파일에서 이 파일을 모듈로 불러올 때는 `main()` 함수가 자동으로 실행되지 않도록 합니다.

`def` 나 `main` 함수는 우리가 직접 틀을 잡고 짤으며 BMI처럼 코드 안에 복잡한 계산하는 부분은 ChatGPT에게 물어보았다.

문제 해결 과정:

사실 이번 코딩을 하면서 100줄이 넘어가다 보니 많은 어려움을 겪었는데 그럴때마다 해결하기 위해 인터넷에도 찾아보고 gpt도 써봤습니다. 특히 BMI계산이나 칼로리 계산 부분에서 gpt에게 많은 도움을 받았습니다.

```
def get_ingredients_and_calories():
    total_calories = 0
    ingredients = {}

    while True:
        ingredient = input("재료를 입력하세요 (끝내려면 '끝' 입력): ").strip().lower()
        if ingredient == '끝':
            break
        amount = input(f"{ingredient}의 양을 입력하세요 (g): ")

        try:
            amount = float(amount)
        except ValueError:
            print("잘못된 양입니다. 숫자를 입력해주세요.")
            continue

        if ingredient in calories_data:
            calories = (calories_data[ingredient] * amount) / 100
            total_calories += calories
            ingredients[ingredient] = (amount, calories)
        else:
            print("이 재료는 데이터베이스에 없습니다.")

    return ingredients, total_calories
```

특히 이부분

수정 사항:

1. **소문자 변환:** `ingredient = input().strip().lower()` 로 입력된 재료를 소문자로 변환하고 앞 뒤 공백을 제거함으로써, 사용자가 입력한 재료와 `calories_data` 의 키 값이 일치하도록 했습니다. 예를 들어, `소고기` 와 `브로콜리` 가 대소문자나 공백 차이로 인식되지 않던 문제를 해결했습니다.

이제 입력한 재료가 제대로 데이터베이스에서 인식될 것입니다!

그밖에 공백이나 대소문자 문제로 인식이 안됐던것도 이런식으로 해결해 보았습니다.

변경 및 추가 사항:

1. **에러 처리 추가:** `get_user_info()` 함수와 `get_ingredients_and_calories()` 함수에서 잘못된 입력에 대한 예외 처리를 추가했습니다.
2. **JSON 저장 기능 추가:** 사용자 키와 체중을 JSON 파일로 저장하는 기능을 추가했습니다.
3. **데이터 유효성 검사:** 재료의 양 입력 시 숫자가 아닌 값이 들어오면 오류 메시지를 출력하도록 했습니다.

이제 더 견고한 시스템이 되었으니 실행해보세요!

JSON 저장 기능 추가도 배워 보았습니다.

원래 처음에 짰던 코드는 이랬습니다:

```
import json
calories_data = {
    '사과': 52,
    '닭가슴살': 165,
    '브로콜리': 34,
    '밥': 130,
    '계란': 155,
}

def get_user_info():
    height = float(input("키를 입력하세요 (cm): ")) / 100
    weight = float(input("체중을 입력하세요 (kg): "))
    return height, weight

def calculate_bmi(height, weight):
    return weight / (height ** 2)

def get_ingredients_and_calories():
    total_calories = 0
    ingredients = {}

    while True:
        ingredient = input("재료를 입력하세요 (끝내려면 끝 입력): ")
        if ingredient == '끝':
            break
        amount = float(input(f"{ingredient}의 양을 입력하세요 (g): "))
```

```

        if ingredient in calories_data:
            calories = (calories_data[ingredient] * amount) / 100
            total_calories += calories
            ingredients[ingredient] = (amount, calories)
        else:
            print("이 재료는 데이터베이스에 없습니다.")

    return ingredients, total_calories

def recommend_recipes(ingredients):
    print("\n추천 레시피:")
    for ingredient, (amount, calories) in ingredients.items():
        print(f"{ingredient}: {amount}g - {calories:.2f} kcal")

def main():
    print("자동화된 식사 추천 시스템에 오신 것을 환영합니다!")

    height, weight = get_user_info()
    bmi = calculate_bmi(height, weight)
    print(f"당신의 BMI는 {bmi:.2f}입니다.")

    ingredients, total_calories = get_ingredients_and_calories()
    print(f"총 칼로리: {total_calories:.2f} kcal")

    recommend_recipes(ingredients)

if __name__ == "__main__":
    main()

```

여기서 저런 도움들을 받아 지금 코드가 되었습니다.

여기서 끝이지만 마치기 전에 저희가 왜 이걸 골랐는지도 말씀드리겠습니다. 저희는 매일 뭘 먹을지 고민하는 현대인들에 고민을 줄여주기 위해 실제로 유용한 냉장고에 뭐가 있는지 입력 받고 더 나아가 개인에게 맞는 레시피를 추천해줄수 있는 코드를 짜보았습니다. 사실 이 코드에 가능성은 무궁무진하다고 생각합니다 레시피 추천을 넘어 더 세세한 부분까지 구현할수도 있겠죠 나중엔. 이상으로 마치겠습니다. 긴 리포트 읽어 주셔서 감사드립니다.