

## **FOOD MACRO LOOKUP — PROJECT REPORT**

### **1. Introduction**

This project allows users to quickly find the macronutrients of common foods.

It is built using Python and runs a simple Streamlit interface. The main purpose is to calculate protein, fat, and carbohydrates for any selected food.

### **2. Objective**

- Allow fast lookup of food nutrition
- Show protein, fat, and carbohydrates per 100g
- Allow users to enter custom serving size
- Provide downloadable CSV output

### **3. Technologies Used**

- Python
- Streamlit
- Pandas

### **4. Working**

1. User enters a food name
2. Fuzzy-matching algorithm finds the closest food
3. App retrieves macros from built-in dictionary
4. Values shown per 100g and scaled per serving
5. CSV report can be downloaded

### **5. Result**

The app runs locally, instantly gives nutrition values, and works across devices.

### **6. Conclusion**

The project demonstrates a functional and easy nutrition lookup tool. It can be extended with full nutrition databases in the future.

## 7. Screenshots

The screenshot shows a Jupyter Notebook interface with a Python script named 'app.py' open. The code defines a 'MACRO\_DB' dictionary containing nutritional information for various foods. It includes a 'find\_best\_match' function that uses the Levenshtein distance to find the closest match for a given query.

```
File Edit Selection View Go Run ... < - > Welcome app.py Foodproject Explorer OPENED EDITORS foodproject > venv app.py
```

```
app.py (1) find_best_match
1 import rextream as rt
2 import pandas as pd
3 from difflib import get_close_matches
4
5 # Small built-in macro database (values are grams per 100 g)
6
7 MACRO_DB = {
8     "apple": {"protein": 0.3, "fat": 0.2, "carbs": 14.0},
9     "banana": {"protein": 1.1, "fat": 0.3, "carbs": 23.0},
10    "white rice (cooked)": {"protein": 2.7, "fat": 0.3, "carbs": 28.0},
11    "brown rice (cooked)": {"protein": 2.6, "fat": 1.0, "carbs": 23.0},
12    "boiled potato": {"protein": 2.0, "fat": 0.1, "carbs": 17.0},
13    "chicken breast (cooked)": {"protein": 23.0, "fat": 3.0, "carbs": 0.0},
14    "steak (cooked)": {"protein": 25.0, "fat": 13.0, "carbs": 0.0},
15    "egg (whole, cooked)": {"protein": 13.0, "fat": 11.0, "carbs": 1.0},
16    "cheddar cheese": {"protein": 25.0, "fat": 33.0, "carbs": 1.3},
17    "pizza (average)": {"protein": 11.0, "fat": 10.0, "carbs": 26.0},
18    "peanut butter": {"protein": 25.0, "fat": 50.0, "carbs": 20.0},
19    "pancake (one cheese pancake)": {"protein": 1.0, "fat": 1.0, "carbs": 1.2},
20    "tuna pizza": {"protein": 2.4, "fat": 11.0, "carbs": 42.0},
21    "ice cream (vanilla)": {"protein": 3.0, "fat": 11.0, "carbs": 24.0},
22    "fried rice": {"protein": 6.0, "fat": 8.0, "carbs": 28.0}
23 }
24
25 def find_best_match(query, choices, cutoff=0.5, n=3):
26     q = query.strip().lower()
27     if q in choices:
28         return q, 1.0
29     else:
30         possible = get_close_matches(q, choices, n=n, cutoff=cutoff)
```

