```
import pandas as pd
import numpy as np
a=pd.read_csv("/content/drive/MyDrive/minor-/food_data_1_missing_values_real.csv")
b=pd.read_csv("/content/drive/MyDrive/minor-/FOOD-DATA-GROUP2.csv")
c=pd.read_csv("/content/drive/MyDrive/minor-/FOOD-DATA-GROUP3.csv")
d=pd.read_csv("/content/drive/MyDrive/minor-/food_data_4_missing_real.csv")
e=pd.read_csv("/content/drive/MyDrive/minor-/FOOD-DATA-GROUP5.csv")

#Merge 5 dataset
data = pd.concat([a,b,c,d,e], axis=0)
print(data)
```

```
     Unnamed: 0.1  Unnamed: 0                             food  \
0               0           0                     cream cheese
1               1           1                neufchatel cheese
2               2           2  requeijao cremoso light catupiry
3               3           3                   ricotta cheese
4               4           4             cream cheese low fat
..            ...         ...                              ...
717           717         717                         jews ear
718           718         718                  enoki mushrooms
719           719         719                  morel mushrooms
720           720         720          portabella mushrooms raw
721           721         721                   oyster mushroom

     Caloric Value     Fat  Saturated Fats  Monounsaturated Fats  \
0             51.0   5.000           2.900                 1.300
1            215.0  19.400          10.900                 4.900
2             49.0     NaN           2.300                 0.900
3             30.0   2.000           1.300                 0.500
4             30.0   2.300           1.400                 0.600
..             ...     ...             ...                   ...
717           25.0   0.095           0.000                 0.000
718            1.0   0.099           0.027                 0.000
719            4.0   0.070           0.056                 0.031
720           19.0   0.300           0.036                 0.016
721            5.0   0.035           0.016                 0.039

     Polyunsaturated Fats  Carbohydrates  Sugars  ...  Calcium  Copper   Iron  \
0                   0.200            0.8   0.500  ...    0.008  14.100  0.082
1                   0.800            3.1   2.700  ...   99.500   0.034  0.100
2                   0.000            0.9   3.400  ...    0.000   0.000  0.000
3                   0.002            1.5   0.091  ...    0.097  41.200  0.097
4                   0.042            1.2   0.900  ...   22.200   0.072  0.008
..                    ...            ...     ...  ...      ...     ...    ...
717                 0.000            6.7   0.000  ...   15.800   0.400  0.600
718                 0.010            0.2   0.034  ...    0.000   0.000  0.099
719                 0.007            0.7   0.096  ...    0.000   5.500  0.056
720                 0.100            3.3   2.200  ...    2.600   0.200  0.300
721                 0.099            0.9   0.200  ...    0.000   0.500  0.008
```

```
     Magnesium  Manganese  Phosphorus  Potassium  Selenium   Zinc  \
0        0.027      1.300       0.091       15.5    19.100  0.039
1        8.500      0.088     117.300      129.2     0.054  0.700
2        0.000      0.000       0.000        0.0     0.000  0.000
3        0.096      4.000       0.024       30.8    43.800  0.035
4        1.200      0.098      22.800       37.1     0.034  0.053
..         ...        ...         ...        ...       ...    ...
717     24.800      0.040      13.900       42.6     0.034  0.700
718      0.034      0.500       0.015        3.2    10.800  0.096
719      1.600      2.500       0.060       25.0    53.000  0.069
720      0.000      0.092      92.900      313.0     0.015  0.500
721      0.200      2.700       0.055       18.0    63.000  0.048

     Nutrition Density
0                7.070
1              130.100
2                5.400
3                5.196
4               27.007
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
 #Step 1: Remove redundant columns
cleaned_data = data.drop(columns=['Unnamed: 0.1', 'Unnamed: 0'])
print(cleaned_data)
```

```
                               food  Caloric Value     Fat  Saturated Fats  \
0                      cream cheese           51.0   5.000           2.900
1                 neufchatel cheese          215.0  19.400          10.900
2    requeijao cremoso light catupiry          49.0     NaN           2.300
3                    ricotta cheese           30.0   2.000           1.300
4                cream cheese low fat          30.0   2.300           1.400
..                              ...            ...     ...             ...
717                        jews ear           25.0   0.095           0.000
718                 enoki mushrooms            1.0   0.099           0.027
719                 morel mushrooms            4.0   0.070           0.056
720         portabella mushrooms raw          19.0   0.300           0.036
721                  oyster mushroom           5.0   0.035           0.016

     Monounsaturated Fats  Polyunsaturated Fats  Carbohydrates  Sugars  \
0                   1.300                 0.200            0.8   0.500
1                   4.900                 0.800            3.1   2.700
2                   0.900                 0.000            0.9   3.400
3                   0.500                 0.002            1.5   0.091
4                   0.600                 0.042            1.2   0.900
..                    ...                   ...            ...     ...
717                 0.000                 0.000            6.7   0.000
718                 0.000                 0.010            0.2   0.034
719                 0.031                 0.007            0.7   0.096
720                 0.016                 0.100            3.3   2.200
721                 0.039                 0.099            0.9   0.200

     Protein  Dietary Fiber  ...  Calcium  Copper   Iron  Magnesium  \
0      0.900          0.000  ...    0.008  14.100  0.082      0.027
```

```
1      7.800          0.000  ...   99.500   0.034  0.100      8.500
2      0.800          0.100  ...    0.000   0.000  0.000      0.000
3      1.500          0.000  ...    0.097  41.200  0.097      0.096
4      1.200          0.000  ...   22.200   0.072  0.008      1.200
..       ...            ...  ...      ...     ...    ...        ...
717    0.500          0.000  ...   15.800   0.400  0.600     24.800
718    0.062          0.089  ...    0.000   0.000  0.099      0.034
719    0.400          0.400  ...    0.000   5.500  0.056      1.600
720    1.800          1.100  ...    2.600   0.200  0.300      0.000
721    0.500          0.300  ...    0.000   0.500  0.008      0.200

     Manganese  Phosphorus  Potassium  Selenium   Zinc  Nutrition Density
0        1.300       0.091       15.5    19.100  0.039              7.070
1        0.088     117.300      129.2     0.054  0.700            130.100
2        0.000       0.000        0.0     0.000  0.000              5.400
3        4.000       0.024       30.8    43.800  0.035              5.196
4        0.098      22.800       37.1     0.034  0.053             27.007
..         ...         ...        ...       ...    ...                ...
717      0.040      13.900       42.6     0.034  0.700             24.249
718      0.500       0.015        3.2    10.800  0.096              0.433
719      2.500       0.060       25.0    53.000  0.069              1.727
720      0.092      92.900      313.0     0.015  0.500              9.400
721      2.700       0.055       18.0    63.000  0.048              1.804

[2395 rows x 35 columns]
```

```python
# Step 2: Drop duplicates
cleaned_data = cleaned_data.drop_duplicates()
print("\nData after removing duplicates:\n", cleaned_data)
```

```
Data after removing duplicates:
                                  food  Caloric Value     Fat  Saturated Fats  \
0                        cream cheese           51.0   5.000           2.900
1                   neufchatel cheese          215.0  19.400          10.900
2    requeijao cremoso light catupiry           49.0     NaN           2.300
3                      ricotta cheese           30.0   2.000           1.300
4                 cream cheese low fat          30.0   2.300           1.400
..                                ...            ...     ...             ...
717                          jews ear           25.0   0.095           0.000
718                   enoki mushrooms            1.0   0.099           0.027
719                   morel mushrooms            4.0   0.070           0.056
720           portabella mushrooms raw          19.0   0.300           0.036
721                   oyster mushroom            5.0   0.035           0.016

     Monounsaturated Fats  Polyunsaturated Fats  Carbohydrates  Sugars  \
0                   1.300                 0.200            0.8   0.500
1                   4.900                 0.800            3.1   2.700
2                   0.900                 0.000            0.9   3.400
3                   0.500                 0.002            1.5   0.091
4                   0.600                 0.042            1.2   0.900
..                    ...                   ...            ...     ...
717                 0.000                 0.000            6.7   0.000
718                 0.000                 0.010            0.2   0.034
719                 0.031                 0.007            0.7   0.096
720                 0.016                 0.100            3.3   2.200
```

```
721                0.039                 0.099           0.9   0.200

     Protein  Dietary Fiber  ...  Calcium  Copper   Iron  Magnesium  \
0      0.900          0.000  ...    0.008  14.100  0.082      0.027
1      7.800          0.000  ...   99.500   0.034  0.100      8.500
2      0.800          0.100  ...    0.000   0.000  0.000      0.000
3      1.500          0.000  ...    0.097  41.200  0.097      0.096
4      1.200          0.000  ...   22.200   0.072  0.008      1.200
..       ...            ...  ...      ...     ...    ...        ...
717    0.500          0.000  ...   15.800   0.400  0.600     24.800
718    0.062          0.089  ...    0.000   0.000  0.099      0.034
719    0.400          0.400  ...    0.000   5.500  0.056      1.600
720    1.800          1.100  ...    2.600   0.200  0.300      0.000
721    0.500          0.300  ...    0.000   0.500  0.008      0.200

     Manganese  Phosphorus  Potassium  Selenium   Zinc  Nutrition Density
0        1.300       0.091       15.5    19.100  0.039              7.070
1        0.088     117.300      129.2     0.054  0.700            130.100
2        0.000       0.000        0.0     0.000  0.000              5.400
3        4.000       0.024       30.8    43.800  0.035              5.196
4        0.098      22.800       37.1     0.034  0.053             27.007
..         ...         ...        ...       ...    ...                ...
717      0.040      13.900       42.6     0.034  0.700             24.249
718      0.500       0.015        3.2    10.800  0.096              0.433
719      2.500       0.060       25.0    53.000  0.069              1.727
720      0.092      92.900      313.0     0.015  0.500              9.400
721      2.700       0.055       18.0    63.000  0.048              1.804

[2395 rows x 35 columns]
```

```
# Step 3: Check for missing values
missing_values = cleaned_data.isnull().sum()
print("\nMissing Values Summary:\n", missing_values)
```

```
Missing Values Summary:
 food                    0
Caloric Value          27
Fat                    55
Saturated Fats          0
Monounsaturated Fats    0
Polyunsaturated Fats    0
Carbohydrates           0
Sugars                 11
Protein                 0
Dietary Fiber           0
Cholesterol             0
Sodium                  0
Water                   0
Vitamin A               0
Vitamin B1              0
Vitamin B11             0
Vitamin B12             0
Vitamin B2              0
Vitamin B3              0
Vitamin B5             38
Vitamin B6              0
```

```
Vitamin C             0
Vitamin D             0
Vitamin E             0
Vitamin K             0
Calcium               0
Copper                0
Iron                 11
Magnesium             0
Manganese             0
Phosphorus            0
Potassium             0
Selenium              0
Zinc                  0
Nutrition Density     0
dtype: int64
```

```
cleaned_data['Caloric Value'].fillna(cleaned_data['Caloric Value'].mean(),inplace=True) #hanling missing values
```

Show hidden output

```
cleaned_data['Fat'].fillna(cleaned_data['Fat'].mean(),inplace=True) #hanling missing values
```

Show hidden output

```
cleaned_data['Sugars'].fillna(cleaned_data['Sugars'].mean(),inplace=True) #hanling missing values
```

Show hidden output

```
cleaned_data['Vitamin B5'].fillna(cleaned_data['Vitamin B5'].mean(),inplace=True) #hanling missing values
```

Show hidden output

```
cleaned_data['Iron'].fillna(cleaned_data['Iron'].mean(),inplace=True) #hanling missing values
```

Show hidden output

```
#Check for missing values again
missing_values = cleaned_data.isnull().sum()
print("\nMissing Values Summary:\n", missing_values)
```

```
Missing Values Summary:
 food                  0
Caloric Value        0
Fat                  0
Saturated Fats       0
Monounsaturated Fats 0
Polyunsaturated Fats 0
Carbohydrates        0
Sugars               0
```

```
Protein                 0
Dietary Fiber           0
Cholesterol             0
Sodium                  0
Water                   0
Vitamin A               0
Vitamin B1              0
Vitamin B11             0
Vitamin B12             0
Vitamin B2              0
Vitamin B3              0
Vitamin B5              0
Vitamin B6              0
Vitamin C               0
Vitamin D               0
Vitamin E               0
Vitamin K               0
Calcium                 0
Copper                  0
Iron                    0
Magnesium               0
Manganese               0
Phosphorus              0
Potassium               0
Selenium                0
Zinc                    0
Nutrition Density       0
dtype: int64
```

```
cleaned_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2395 entries, 0 to 721
Data columns (total 35 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   food                  2395 non-null   object
 1   Caloric Value         2395 non-null   float64
 2   Fat                   2395 non-null   float64
 3   Saturated Fats        2395 non-null   float64
 4   Monounsaturated Fats  2395 non-null   float64
 5   Polyunsaturated Fats  2395 non-null   float64
 6   Carbohydrates         2395 non-null   float64
 7   Sugars                2395 non-null   float64
 8   Protein               2395 non-null   float64
 9   Dietary Fiber         2395 non-null   float64
 10  Cholesterol           2395 non-null   float64
 11  Sodium                2395 non-null   float64
 12  Water                 2395 non-null   float64
 13  Vitamin A             2395 non-null   float64
 14  Vitamin B1            2395 non-null   float64
 15  Vitamin B11           2395 non-null   float64
 16  Vitamin B12           2395 non-null   float64
 17  Vitamin B2            2395 non-null   float64
 18  Vitamin B3            2395 non-null   float64
 19  Vitamin B5            2395 non-null   float64
 20  Vitamin B6            2395 non-null   float64
```

```
 21  Vitamin C             2395 non-null   float64
 22  Vitamin D             2395 non-null   float64
 23  Vitamin E             2395 non-null   float64
 24  Vitamin K             2395 non-null   float64
 25  Calcium               2395 non-null   float64
 26  Copper                2395 non-null   float64
 27  Iron                  2395 non-null   float64
 28  Magnesium             2395 non-null   float64
 29  Manganese             2395 non-null   float64
 30  Phosphorus            2395 non-null   float64
 31  Potassium             2395 non-null   float64
 32  Selenium              2395 non-null   float64
 33  Zinc                  2395 non-null   float64
 34  Nutrition Density     2395 non-null   float64
dtypes: float64(34), object(1)
memory usage: 673.6+ KB
```

```
# Step 5: Standardize column names (convert to lowercase and replace spaces with underscores)
cleaned_data.columns = cleaned_data.columns.str.capitalize().str.replace(' ', '_')
#cleaned_data['Food'] = cleaned_data['Food'].str.capitalize()
print(cleaned_data.head())
```

```
                               Food  Caloric_value        Fat  Saturated_fats  \
0                      cream cheese           51.0   5.000000             2.9
1                 neufchatel cheese          215.0  19.400000            10.9
2  requeijao cremoso light catupiry           49.0  10.130935             2.3
3                    ricotta cheese           30.0   2.000000             1.3
4              cream cheese low fat           30.0   2.300000             1.4

   Monounsaturated_fats  Polyunsaturated_fats  Carbohydrates  Sugars  Protein  \
0                   1.3                 0.200            0.8   0.500      0.9
1                   4.9                 0.800            3.1   2.700      7.8
2                   0.9                 0.000            0.9   3.400      0.8
3                   0.5                 0.002            1.5   0.091      1.5
4                   0.6                 0.042            1.2   0.900      1.2

   Dietary_fiber  ...  Calcium  Copper   Iron  Magnesium  Manganese  \
0            0.0  ...    0.008  14.100  0.082      0.027      1.300
1            0.0  ...   99.500   0.034  0.100      8.500      0.088
2            0.1  ...    0.000   0.000  0.000      0.000      0.000
3            0.0  ...    0.097  41.200  0.097      0.096      4.000
4            0.0  ...   22.200   0.072  0.008      1.200      0.098

   Phosphorus  Potassium  Selenium   Zinc  Nutrition_density
0       0.091       15.5    19.100  0.039              7.070
1     117.300      129.2     0.054  0.700            130.100
2       0.000        0.0     0.000  0.000              5.400
3       0.024       30.8    43.800  0.035              5.196
4      22.800       37.1     0.034  0.053             27.007

[5 rows x 35 columns]
```

```
# Save the cleaned data to a new CSV file
#cleaned_data.to_csv(r"//content/drive/MyDrive/minor-/new.csv", index=False)
```

```
import pandas as pd


data = pd.read_csv("/content/drive/MyDrive/minor-/new.csv")


non_veg_keywords = [
    'chicken','meat','fish', 'beef', 'mutton', 'egg', 'pork', 'lamb', 'shrimp',
    'turkey', 'crab', 'duck', 'bacon', 'ham', 'sausage', 'salami', 'lobster',
    'prawns', 'squid', 'octopus', 'veal', 'goat', 'quail', 'venison', 'foie gras',
    'oyster', 'scallop', 'clam', 'anchovy', 'tuna', 'salmon', 'sardine', 'caviar',
    'goose', 'kangaroo', 'boar', 'pepperoni', 'meatball', 'kebab', 'pastrami',
    'hot dog', 'prosciutto', 'jerky', 'escargot', 'tripe', 'liver', 'kidney',
    'giblets', 'bison', 'buffalo', 'rabbit', 'elk', 'frog', 'snail', 'conch',
    'cod', 'herring', 'halibut', 'swordfish', 'snapper', 'bass', 'flounder',
    'tilapia', 'mussels', 'calamari', 'eel', 'shark', 'alligator', 'ostrich',
    'reindeer', 'pheasant', 'squab', 'rattlesnake', 'turtle', 'roe', 'sea bass',
    'monkfish', 'pike', 'catfish', 'walleye', 'mahi mahi', 'grouper', 'haddock',
    'chorizo', 'salpicao', 'mortadella', 'serrano', 'chicken wings', 'ribs',
    'ox', 'trout', 'carp', 'partridge', 'moose', 'emu', 'iguana', 'wild boar',
    'black pudding', 'blood sausage', 'sweetbreads', 'bone marrow', 'head cheese'
]


def categorize_food(food_name):

    if isinstance(food_name, str):
        food_name_lower = food_name.lower()


        if any(keyword in food_name_lower for keyword in non_veg_keywords):
            return 'Non-Vegetarian'



    return 'Vegetarian'

data['Category'] = data['Food'].apply(categorize_food)

# Display the updated dataset
print(data.head())
```

```
                              Food  Caloric_value        Fat  Saturated_fats  \
0                     cream cheese           51.0   5.000000             2.9
1                 neufchatel cheese         215.0  19.400000            10.9
2  requeijao cremoso light catupiry          49.0  10.130935             2.3
3                   ricotta cheese           30.0   2.000000             1.3
4              cream cheese low fat          30.0   2.300000             1.4

   Monounsaturated_fats  Polyunsaturated_fats  Carbohydrates  Sugars  Protein  \
0                   1.3                 0.200            0.8   0.500      0.9
1                   4.9                 0.800            3.1   2.700      7.8
2                   0.9                 0.000            0.9   3.400      0.8
```

```
3                  0.5                0.002           1.5   0.091      1.5
4                  0.6                0.042           1.2   0.900      1.2

   Dietary_fiber  ...  Copper   Iron  Magnesium  Manganese  Phosphorus  \
0            0.0  ...  14.100  0.082      0.027      1.300       0.091
1            0.0  ...   0.034  0.100      8.500      0.088     117.300
2            0.1  ...   0.000  0.000      0.000      0.000       0.000
3            0.0  ...  41.200  0.097      0.096      4.000       0.024
4            0.0  ...   0.072  0.008      1.200      0.098      22.800

   Potassium  Selenium   Zinc  Nutrition_density    Category
0       15.5    19.100  0.039              7.070  Vegetarian
1      129.2     0.054  0.700            130.100  Vegetarian
2        0.0     0.000  0.000              5.400  Vegetarian
3       30.8    43.800  0.035              5.196  Vegetarian
4       37.1     0.034  0.053             27.007  Vegetarian

[5 rows x 36 columns]
```

```
import matplotlib.pyplot as plt
def get_user_input():
    print("Welcome to the Food Recommendation System!")

    has_disease = input("Do you have any special disease? Press 1 for Yes, Press 2 for No: ").strip()


    diseases = [
        "Diabetes", "Hypertension (High Blood Pressure)", "Obesity", "Heart Disease (Cardiovascular Disease)",
        "Asthma", "Chronic Obstructive Pulmonary Disease (COPD)", "Cancer", "Stroke", "Alzheimer's Disease",
        "Parkinson's Disease", "Arthritis (Osteoarthritis/Rheumatoid Arthritis)", "Chronic Kidney Disease",
        "Hepatitis (A, B, C)", "HIV/AIDS", "Influenza (Flu)", "Pneumonia", "Tuberculosis (TB)", "Malaria",
        "Dementia", "Irritable Bowel Syndrome (IBS)", "Epilepsy", "Migraine", "Gastritis", "Celiac Disease",
        "Lupus", "Thyroid Disorders (Hypothyroidism/Hyperthyroidism)", "Gout", "Multiple Sclerosis (MS)",
        "Psoriasis", "Eczema"
    ]

    disease = None
    if has_disease == '1':
        print("\nSelect your disease from the list below:")
        for idx, d in enumerate(diseases, start=1):
            print(f"{idx}: {d}")

        disease_choice = int(input("Enter the number corresponding to your disease: ").strip())
        if 1 <= disease_choice <= len(diseases):
            disease = diseases[disease_choice - 1]
        else:
            print("Invalid choice, proceeding without specific disease.")


    vegetarian = input("\nAre you vegetarian? (yes/no): ").strip().lower() == 'yes'


    nutrients = nutrients = [
    "Caloric_value", "Fat", "Saturated_fats", "Monounsaturated_fats", "Polyunsaturated_fats",
```

```
        "Carbohydrates", "Sugars", "Protein", "Dietary_fiber", "Cholesterol", "Sodium", "Water",
        "Vitamin_a", "Vitamin_b1", "Vitamin_b11", "Vitamin_b12", "Vitamin_b2", "Vitamin_b3",
        "Vitamin_b5", "Vitamin_b6", "Vitamin_c", "Vitamin_d", "Vitamin_e", "Vitamin_k",
        "Calcium", "Copper", "Iron", "Magnesium", "Manganese", "Phosphorus", "Potassium",
        "Selenium", "Zinc"
    ]


    print("\nSelect up to 3 nutrient deficiencies from the list below:")
    for idx, n in enumerate(nutrients, start=1):
        print(f"{idx}: {n}")


    deficiencies = []
    while len(deficiencies) < 3:
        try:
            choice = int(input(f"Enter the number corresponding to deficiency {len(deficiencies)+1} (or 0 to stop): ").strip())
            if choice == 0:
                break
            if 1 <= choice <= len(nutrients):
                deficiencies.append(nutrients[choice - 1])
            else:
                print("Invalid choice, try again.")
        except ValueError:
            print("Please enter a valid number.")

    print("\nWhat is your primary health goal? (Choose one)")
    print("Options: muscle gain, weight loss, overall health")
    health_goal = input("Your choice: ").strip().lower()


    user_data = {
        'disease': disease,
        'vegetarian': vegetarian,
        'deficiencies': deficiencies,
        'health_goal': health_goal
    }

    return user_data


user_input = get_user_input()
print("\nUser Input Summary:")
print(f"Disease: {user_input['disease']}")
print(f"Vegetarian: {user_input['vegetarian']}")
print(f"Nutrient Deficiencies: {', '.join(user_input['deficiencies'])}")
print(f"Health Goal: {user_input['health_goal']}")

food_data=data.copy()



def apply_disease_criteria(food_data, disease):
```

```
        disease_conditions = {
            "Diabetes": (food_data['Sugars'] < 5) & (food_data['Carbohydrates'] < 60) & (food_data['Dietary_fiber'] > 5),
            "Hypertension (High Blood Pressure)": (food_data['Sodium'] < 300) & (food_data['Potassium'] > 200),
            "Obesity": (food_data['Caloric_value'] < 400) & (food_data['Dietary_fiber'] > 5) & (food_data['Protein'] > 20),
            "Heart Disease (Cardiovascular Disease)": (food_data['Saturated_fats'] < 3) & (food_data['Cholesterol'] < 50) & (food_data['Sodium'] < 300) & (food_data['Dietary_fiber'] > 5),
            "Asthma": (food_data['Vitamin_c'] > 15) & (food_data['Vitamin_e'] > 2) & (food_data['Saturated_fats'] < 3),
            "Chronic Obstructive Pulmonary Disease (COPD)": (food_data['Dietary_fiber'] < 5) & (food_data['Caloric_value'] > 400),
            "Cancer": (food_data['Vitamin_a'] > 300) & (food_data['Vitamin_c'] > 20) & (food_data['Dietary_fiber'] > 5),
            "Stroke": (food_data['Potassium'] > 300) & (food_data['Sodium'] < 300),
            "Alzheimer's Disease": (food_data['Vitamin_e'] > 2) & (food_data['Saturated_fats'] < 3),
            "Parkinson's Disease": (food_data['Protein'] > 15) & (food_data['Vitamin_b12'] > 2)
        }

        if disease in disease_conditions:
            return food_data.loc[disease_conditions[disease]]
        else:
            return food_data

    disease=user_input['disease']
    filtered_data = apply_disease_criteria(food_data, disease)


    def apply_veg_nonveg_preference(filtered_data, vegetarian_preference):

        if vegetarian_preference:

            filtered_data = filtered_data[filtered_data['Category'].str.lower() == 'vegetarian']
        else:

            filtered_data = filtered_data[filtered_data['Category'].str.lower() == 'non-vegetarian']

        return filtered_data









    vegetarian_preference=user_input['vegetarian']
    filtered_foods = apply_veg_nonveg_preference(filtered_data, vegetarian_preference)






    def apply_nutrient_deficiency(filtered_data, deficiencies):

        deficiency_conditions = {
            "Caloric Value": filtered_data['Caloric_value'] > filtered_data['Caloric_value'].median(),
            "Fat": filtered_data['Fat'] > filtered_data['Fat'].median(),
            "Saturated Fats": filtered_data['Saturated_fats'] > filtered_data['Saturated_fats'].median(),
```

```
        "Monounsaturated Fats": filtered_data['Monounsaturated_fats'] > filtered_data['Monounsaturated_fats'].median(),
        "Polyunsaturated Fats": filtered_data['Polyunsaturated_fats'] > filtered_data['Polyunsaturated_fats'].median(),
        "Carbohydrates": filtered_data['Carbohydrates'] > filtered_data['Carbohydrates'].median(),
        "Sugars": filtered_data['Sugars'] > filtered_data['Sugars'].median(),
        "Protein": filtered_data['Protein'] > filtered_data['Protein'].median(),
        "Dietary Fiber": filtered_data['Dietary_fiber'] > filtered_data['Dietary_fiber'].median(),
        "Cholesterol": filtered_data['Cholesterol'] > filtered_data['Cholesterol'].median(),
        "Sodium": filtered_data['Sodium'] > filtered_data['Sodium'].median(),
        "Water": filtered_data['Water'] > filtered_data['Water'].median(),
        "Vitamin A": filtered_data['Vitamin_a'] > filtered_data['Vitamin_a'].median(),
        "Vitamin B1": filtered_data['Vitamin_b1'] > filtered_data['Vitamin_b1'].median(),
        "Vitamin B11": filtered_data['Vitamin_b11'] > filtered_data['Vitamin_b11'].median(),
        "Vitamin B12": filtered_data['Vitamin_b12'] > filtered_data['Vitamin_b12'].median(),
        "Vitamin B2": filtered_data['Vitamin_b2'] > filtered_data['Vitamin_b2'].median(),
        "Vitamin B3": filtered_data['Vitamin_b3'] > filtered_data['Vitamin_b3'].median(),
        "Vitamin B5": filtered_data['Vitamin_b5'] > filtered_data['Vitamin_b5'].median(),
        "Vitamin B6": filtered_data['Vitamin_b6'] > filtered_data['Vitamin_b6'].median(),
        "Vitamin C": filtered_data['Vitamin_c'] > filtered_data['Vitamin_c'].median(),
        "Vitamin D": filtered_data['Vitamin_d'] > filtered_data['Vitamin_d'].median(),
        "Vitamin E": filtered_data['Vitamin_e'] > filtered_data['Vitamin_e'].median(),
        "Vitamin K": filtered_data['Vitamin_k'] > filtered_data['Vitamin_k'].median(),
        "Calcium": filtered_data['Calcium'] > filtered_data['Calcium'].median(),
        "Copper": filtered_data['Copper'] > filtered_data['Copper'].median(),
        "Iron": filtered_data['Iron'] > filtered_data['Iron'].median(),
        "Magnesium": filtered_data['Magnesium'] > filtered_data['Magnesium'].median(),
        "Manganese": filtered_data['Manganese'] > filtered_data['Manganese'].median(),
        "Phosphorus": filtered_data['Phosphorus'] > filtered_data['Phosphorus'].median(),
        "Potassium": filtered_data['Potassium'] > filtered_data['Potassium'].median(),
        "Selenium": filtered_data['Selenium'] > filtered_data['Selenium'].median(),
        "Zinc": filtered_data['Zinc'] > filtered_data['Zinc'].median()
    }


    combined_condition = pd.Series([False] * len(filtered_data), index=filtered_data.index)


    for deficiency in deficiencies:
        if deficiency in deficiency_conditions:

            combined_condition |= deficiency_conditions[deficiency]

    return filtered_data[combined_condition]
deficiencies = user_input['deficiencies']
filtered_data = apply_nutrient_deficiency(filtered_data, deficiencies)
health_goal = user_input['health_goal']

def apply_health_goal(filtered_data, health_goal):

    health_goal_conditions = {
        "muscle gain": (filtered_data['Protein'] > filtered_data['Protein'].median()) &
                       (filtered_data['Fat'] > filtered_data['Fat'].quantile(0.25)),

        "weight loss": (filtered_data['Fat'] < filtered_data['Fat'].quantile(0.25)) &
                       (filtered_data['Dietary_fiber'] > filtered_data['Dietary_fiber'].median()) &
```

```
                        (filtered_data['Caloric_value'] < filtered_data['Caloric_value'].median()),

        "overall health": (filtered_data['Vitamin_a'] > filtered_data['Vitamin_a'].median()) &
                          (filtered_data['Vitamin_c'] > filtered_data['Vitamin_c'].median()) &
                          (filtered_data['Calcium'] > filtered_data['Calcium'].median()) &
                          (filtered_data['Potassium'] > filtered_data['Potassium'].median()) &
                          (filtered_data['Dietary_fiber'] > filtered_data['Dietary_fiber'].median())

}

    # Apply the condition based on the selected health goal
    if health_goal in health_goal_conditions:
        return filtered_data.loc[health_goal_conditions[health_goal]]
    else:

        print(f"Unknown health goal: {health_goal}. Returning unfiltered data.")
        return filtered_data

filtered_foods = apply_health_goal(filtered_foods, user_input['health_goal']).head(10)



from tabulate import tabulate



table = tabulate(filtered_foods, headers='keys', tablefmt='grid')
print(table)








def plot_food_deficiencies(filtered_foods, deficiencies):

    for deficiency in deficiencies:
        if deficiency in filtered_foods.columns:
            plt.figure(figsize=(10, 6))
            plt.bar(filtered_foods['Food'], filtered_foods[deficiency], color='skyblue')
            plt.title(f'{deficiency} Levels in Recommended Foods')
            plt.xlabel('Food')

            plt.ylabel(f'{deficiency} Level')
            plt.xticks(rotation=45, ha='right')
            plt.tight_layout()
            plt.show()
        else:
            print(f"Deficiency {deficiency} not found in dataset columns.")
```

```
plot_food_deficiencies(filtered_foods, deficiencies)
```

New Section

```
1: Diabetes
2: Hypertension (High Blood Pressure)
3: Obesity
4: Heart Disease (Cardiovascular Disease)
5: Asthma
6: Chronic Obstructive Pulmonary Disease (COPD)
7: Cancer
8: Stroke
9: Alzheimer's Disease
10: Parkinson's Disease
11: Arthritis (Osteoarthritis/Rheumatoid Arthritis)
12: Chronic Kidney Disease
13: Hepatitis (A, B, C)
14: HIV/AIDS
15: Influenza (Flu)
16: Pneumonia
17: Tuberculosis (TB)
18: Malaria
19: Dementia
20: Irritable Bowel Syndrome (IBS)
```