# **README - NTU F&B Search Engine**

## **Overview**

This program helps to recommend canteens in NTU based on three different search criteria:

1. **Keyword-based Search:** Recommends canteens based on specified cuisine keywords. Users can use logical operators such as ‘and’ or ‘or’ to refine their search. For example:
   * "Western and Fries" will match stalls that serve both Western food and fries.
   * "Japanese or Korean" will return stalls that serve either Japanese or Korean cuisine.

The keyword-based search results are now sorted and ranked by how many keywords they match from the user input. For example, if a stall has both “chicken” and “rice” while another has only “chicken,” the former will be ranked higher in the list. This helps users see the most relevant stalls first based on their search.

1. **Price-based Search:** Recommends canteens based on cuisine keywords and an indicated budget.
   * The program first filters stalls based on the keyword search.
   * Then, it further filters the results by removing stalls that exceed the user’s maximum allowable price.

All search outputs (from keyword-based, price-based, and location-based searches) now display as **numbered lists**, making the results easier to read and reference. For instance, instead of showing a block of stall names, the results appear as:

1. North Spine - McRonald's (Matched 2 keywords)

2. Food Court 14 - Burger House (Matched 1 keyword)

1. **Location-based Search:** Recommends the nearest canteens based on a user-selected location on the NTU map.
   * The program uses PyGame to allow the user to click on a map to select a location.
   * The user can specify the number of nearby canteens to display.

## **Key Features & Enhancements**

### **1. Pathlib Module for Dynamic File Paths**

* This allows the program to dynamically locate files inside the folder without requiring manual path changes.

**Example Implementation:**  
DATA\_FILE = str(pathlib.Path(\_\_file\_\_).parent.resolve()) + "\\canteens.xlsx"

This ensures the program can locate canteens.xlsx correctly, regardless of the system’s directory structure.

### **2. Enhanced Keyword Search with AND/OR Logic**

* Enables advanced searching by allowing multiple keywords and logical operators.

**Implementation:**The parse\_keywords() function splits user input based on "and" and "or" operators, returning a structured format.

Example transformation:  
"Western or Burgers and Fries" → [['Western'], ['Burgers', 'Fries']]

To improve the flexibility and accuracy of user input, the keyword parsing function now uses regular expressions to handle cases where "and" or "or" are typed without spaces.  
  
The program then searches for stalls that match at least one "OR" group and all "AND" conditions within each group.

User inputs that contain single spaces between keywords (e.g., "chicken rice western") are now treated as if the user typed “chicken and rice and western”.  
However, multi-word keywords like "mixed rice" are protected and treated as one term, not as "mixed and rice".  
This update enhances input flexibility while preserving full support for complex logical queries using "and" and "or".

### 3. PyGame for User Location Selection

* Provides an interactive way to select a location on the NTU campus map.

Implementation:  
A PyGame interface loads the NTU map (NTUcampus.jpg). The user clicks anywhere on the map to set a location, which is then used to calculate distances to nearby canteens.

### 4. Improved Input Validation & Error Handling

* Ensures valid user input for price, keywords, and number of canteens.
* Handles invalid inputs with appropriate error messages.  
  Example: If a user enters a negative price, the program prompts for valid input instead of crashing.
* Through Fuzzy Keyword Matching for Better User Feedback, the program now suggests similar keywords if no exact match is found. This helps users discover available food types and avoid frustration when making minor typos.

## **Helper Functions**

### **User Input Handling**

* **parse\_keywords(user\_input)** → Parses keywords based on "and" and "or" logic.
* **get\_user\_location\_interface()** → Launches PyGame interface for location selection.
* **confirm\_exit()** → Asks for user confirmation before quitting.

### **Data Processing**

* **load\_canteen\_data(data\_location)** → Reads canteen data from canteens.xlsx and stores keywords, prices, and locations in dictionaries.

### **Search Functions**

* **search\_by\_keyword()** → Filters stalls based on user-provided keywords.
* **search\_by\_price()** → Filters stalls by keyword, then sorts by price.
* **search\_nearest\_canteens()** → Finds the closest canteens to a user-selected location using Euclidean distance.

## **Program Flow**

1. The program **loads the canteen dataset** from canteens.xlsx.
2. The **main menu** is displayed, allowing the user to choose a search method:
   * **Option 1:** Display all canteen data.
   * **Option 2:** Perform a **keyword search** for food stalls.
   * **Option 3:** Perform a **price-based search** with keyword filtering.
   * **Option 4:** Perform a **location-based search** using PyGame.
   * **Option 5:** **Exit** the program.
3. The **search functions process user input**, filter results, and display recommendations.
4. The user can **repeat searches** until they choose to exit the program.

## **Example Use Cases**

### **1. Keyword Search Example**

User input:  
Enter food type: Japanese or Korean  
  
Expected output:  
🍽 Food Stalls Found:  
Canteen 2 - Sushi Express  
Canteen 11 - Seoul Kitchen

### **2. Price-Based Search Example**

User input:  
Enter food type: Chicken Rice   
Enter maximum price: 4.50

Expected output:  
💰 Affordable Food Stalls:  
Canteen 1 - Hainanese Delights: S$3.80  
Canteen 9 - Muthu’s Curry: S$4.20

### **3. Location-Based Search Example**

User selects a location on the NTU map and requests the **3 nearest canteens**.  
 Expected output:  
📍 Nearest Canteens:  
Canteen 13 - 250.75m away  
Canteen 16 - 320.10m away  
Canteen 1 - 415.50m away

## **How to Run the Program**

1. **Ensure you have the required files** in the same directory:
   * NTU\_FnB\_Search.py (this program)
   * canteens.xlsx (dataset)
   * NTUcampus.jpg (map for location-based search)
   * pin.png (marker for selected location)

**Install dependencies** (if not already installed):  
pip install pygame pandas openpyxl

## **Additional Notes**

* This program **only supports NTU food stalls** listed in canteens.xlsx.
* **No real-time updates**: If stall information changes, update canteens.xlsx manually.
* **Supports Windows, Mac, and Linux**, but PyGame may require additional setup on some systems.

## **Acknowledgments**

This project was developed as part of **RE1016 - Engineering Computation** at **Nanyang Technological University (NTU), Assignment 2.**