Code Brief

* By Arnav Tamrakar
* Student ID: 1814996

The main.py is the main file that combines all the children programs to complete the program. It has multiple programs imported as functions and csv files to input data and manipulate it as per need.

1. Writing Menu Data to a CSV File:

The script starts by defining the structure of the menu data and writing it to a CSV file named "menu.csv." The menu data includes information about various food items, such as their names, prices, and categories (e.g., Main, Starter, Dessert). The CSV file is created using the csv module and is written row by row. The menu data is represented as a list of lists, where each inner list corresponds to a menu item.

2. Calculating Average Ratings:

The script imports a module named average\_ratings\_calculator to calculate the average ratings of each item. This module is assumed to contain a function that takes a list of customer ratings for each item and calculates the average rating. The script opens a CSV file named "average\_ratings.csv" and reads the data. It then calculates the average ratings and prints them, rounding to one decimal place.

**The pseudocode of average\_ratings\_calculator.py is:**

Function calculate\_column\_averages(data)

    transposed\_data <- Transpose(data)  
    column\_averages <- Create an empty list

    For each column in transposed\_data:

        column\_sum <- Sum of values in the column  
        average <- column\_sum / Number of values in the column  
        Append average to column\_averages

    Return column\_averages

**The Chart of average\_ratings\_calculator.py is:**

A diagram of a flowchart

Description automatically generated

3. Calculating Total Daily Sales:

The script imports another module named daily\_sales\_calculator to calculate the total daily sales of menu items. Similar to the ratings, this module is assumed to contain a function that computes the total sales for each day. The script reads the data from a CSV file named "daily\_sales.csv," calculates the total sales for each day, and prints the results.

**The pseudocode of daily\_sales\_calculator.py is:**

Function calculate\_total\_sales(data)  
    total\_sales <- Create an empty dictionary  
     
    For each row in data, starting from the second row (skipping the header):  
        day <- Extract the value in the first column (day of the week)  
        sales <- Extract the values in the remaining columns (daily sales)  
         
        total <- 0  # Initialize the total sales for the day  
         
        For each value in sales:  
            If the value is a valid integer:  
                Convert the value to an integer  
                Add the integer value to the total  
                 
        Store the total sales for the day in the total\_sales dictionary using day as the key

Return total\_sales

**The Chart of daily\_sales\_calculator.py is:**

A diagram of sales

Description automatically generated

4. Finding the Most Popular Item:

After calculating the average ratings, the script finds the most popular menu item based on the highest rating. It identifies the item with the highest rating and prints it as the most popular item. However, there is an issue with this part of the code because it sets most\_popular\_item as a list, which should be corrected to specify the most popular item's name.