파이썬 프로그래밍및실습

Developing a food recommendationfunction

Final Report

Date : 23/12/24

Name : 이나경

ID : 231135

1. Introduction

1) Background

Food is something you have to eat every day, but each time you choose a menu,it takes a lot of time to choose a satisfactory menu. This leads to a decrease inbreakfast preparation time, lunch breaks, and work hours while thinking aboutmeals. We need a program that automatically recommends food to solve theproblem of wasting time through eating concerns and to select a more satisfyingmenu.

2) Project goal

The goal is to recommend food menus based on food-specific ratings given bycustomers, reducing the time to think about food during meal times and allowingthem to select more satisfying menus.

3) Differences from existing programs

Existing programs recommend food lists that are already finished and cannot beedited. It is highly likely that foods that do not suit individual preferences willoften be recommended. We have a difference from the existing program in thatby applying a customization method in which customers directly add menus,words they want to leave, allergies, etc., they store only the information theyneed to make customized recommendations for that customer.

2. Functional Requirement

1) **Function 1** Menu Recommendation Function

- Explanation: We recommend the food based on the rating given by thecustomer by the food.

(1) Detailed function 1: Select Category

- Explanation: Allows you to select the category in which you would like to berecommended for food.

(2) Detailed function 2: Save Memo

- Explanation: Save the words you want to leave on the recommended food, suchas recipes, precautions, and allergies.

(3) Detailed function 3: Save Preferences

- Explanation: Leave a preference for the recommended food and allow it to bereflected in the new food recommendations.

2) **Function 2:** Menu editing function

- Explanation: Ability to edit items or information on recommended food lists

(1) Detailed function 1: Add Menu

- Explanation: Add food and leave notes such as preferences.

(2) Detailed function 2: Delete Menu

- Explanation: Delete foods you don't want to be recommended again.

3) **Function 3:** File Storage Function

- Explanation: Upload the newly entered information, such as preferences andnotes, to the file.

3. Implementation

(1) Menu Editing Function - Add Menu

- Input; menus: a dictionary that stores categories,menus, and ratings.

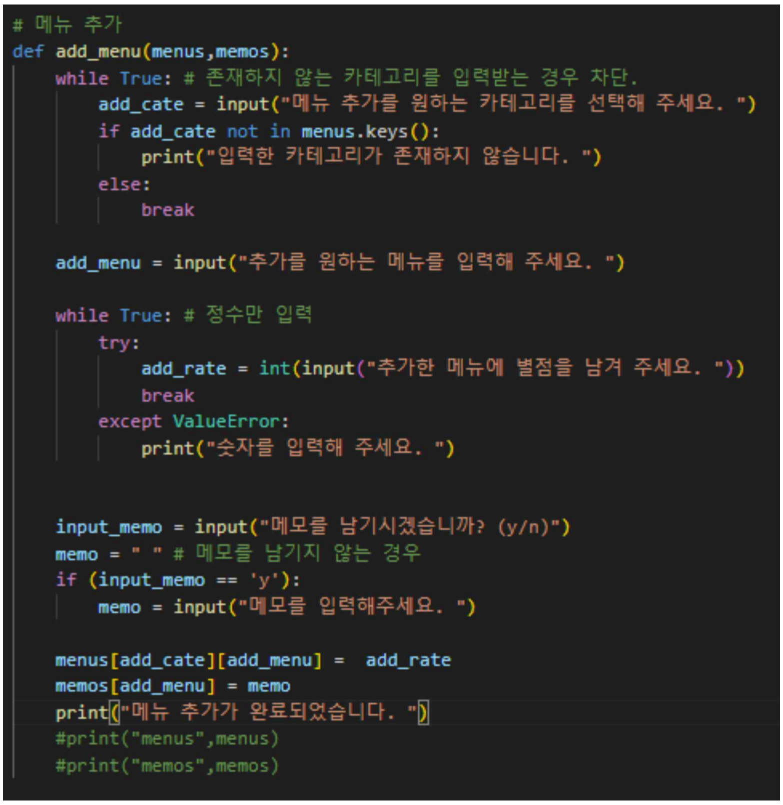
memos: a dictionary that stores menus and memos.

Output; The menu entered by the user and the menu added with rating
Memos with memos corresponding to newly created menu

- Description: First, you get a category input, and if that category is not in the keys of menus, you get it back in. If a category exists, enter the menu and rating you want to add and save additional menus and ratings in menus and memos.

- What you've learned: Repeat statement, condition question, function, module, exception

- Code Screenshot



```python
# 메뉴 추가
def add_menu(menus,memos):
    while True: # 존재하지 않는 카테고리를 입력받는 경우 차단.
        add_cate = input("메뉴 추가를 원하는 카테고리를 선택해 주세요. ")
        if add_cate not in menus.keys():
            print("입력한 카테고리가 존재하지 않습니다. ")
        else:
            break

    add_menu = input("추가를 원하는 메뉴를 입력해 주세요. ")

    while True: # 정수만 입력
        try:
            add_rate = int(input("추가한 메뉴에 별점을 남겨 주세요. "))
            break
        except ValueError:
            print("숫자를 입력해 주세요. ")


    input_memo = input("메모를 남기시겠습니까? (y/n)")
    memo = " " # 메모를 남기지 않는 경우
    if (input_memo == 'y'):
        memo = input("메모를 입력해주세요. ")

    menus[add_cate][add_menu] =  add_rate
    memos[add_menu] = memo
    print("메뉴 추가가 완료되었습니다. ")
    #print("menus",menus)
    #print("memos",memos)
```

(2) Menu Editing Function - Delete Menu

- Input; menus: a dictionary that stores categories,menus, and ratings.

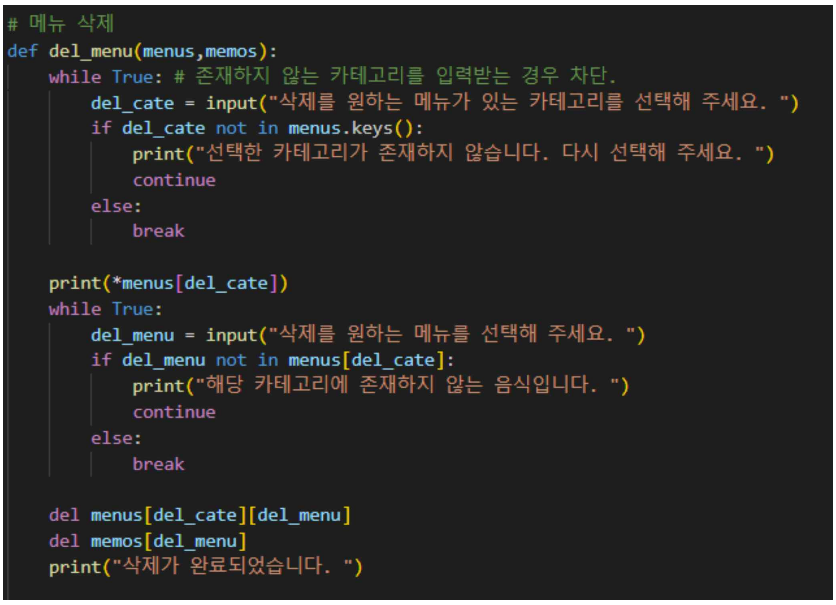
memos: a dictionary that stores menus and memos.

Output: Menus, where the selected menu is deleted, Memo from the selected menu also disappears.

- Description: Enter the category with the menu you want to delete first, and proceed with the deletion only if the category exists in the keys of the menu. Similarly, configure the deletion to proceed only when an incoming menu exists.

- What you've learned: Repeat statement, conditional statement, function, module

- Code Screenshot



```
# 메뉴 삭제
def del_menu(menus,memos):
    while True: # 존재하지 않는 카테고리를 입력받는 경우 차단.
        del_cate = input("삭제를 원하는 메뉴가 있는 카테고리를 선택해 주세요. ")
        if del_cate not in menus.keys():
            print("선택한 카테고리가 존재하지 않습니다. 다시 선택해 주세요. ")
            continue
        else:
            break

    print(*menus[del_cate])
    while True:
        del_menu = input("삭제를 원하는 메뉴를 선택해 주세요. ")
        if del_menu not in menus[del_cate]:
            print("해당 카테고리에 존재하지 않는 음식입니다. ")
            continue
        else:
            break

    del menus[del_cate][del_menu]
    del memos[del_menu]
    print("삭제가 완료되었습니다. ")
```

(3) File Storage Function

- Input; menus: a dictionary that stores categories,menus, and ratings.

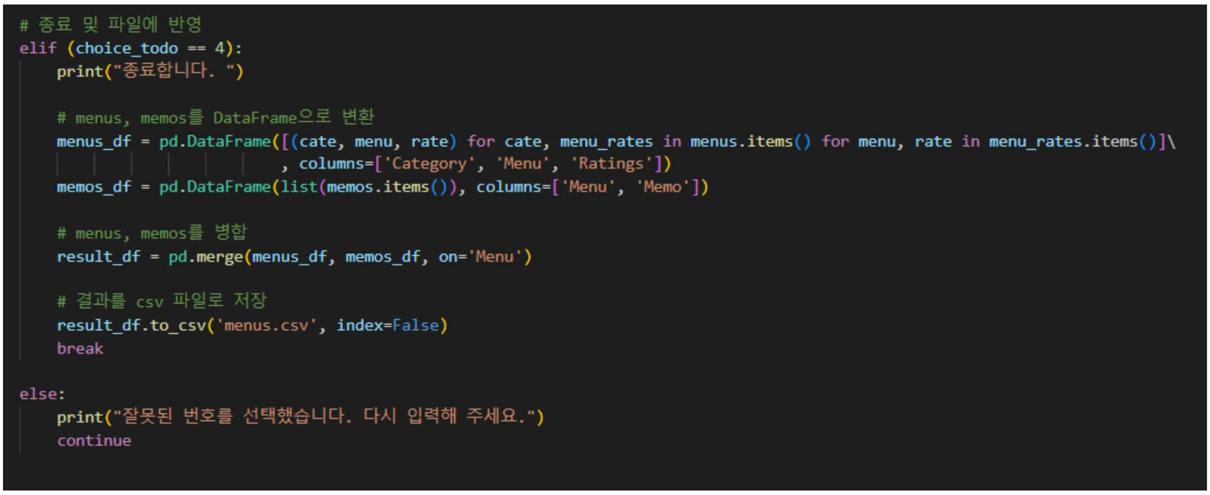
memos: a dictionary that stores menus and memos.

Output: Save to a file reflecting the facts you have added or deleted, And the information you've modified your ratings and memo on your recommendation. Returns the file where the results are stored.

- Description: Create a data frame by extracting Category, Menu, and Rats information from each item in menus. Create another data frame by extracting Menu, Memo information from each item in Memos. Merge the two data frames and save the final data frame as a csv file.

- What you've learned: pandas, Repeat statement

- Code Screenshot



```
# 종료 및 파일에 반영
elif (choice_todo == 4):
    print("종료합니다. ")

    # menus, memos를 DataFrame으로 변환
    menus_df = pd.DataFrame([(cate, menu, rate) for cate, menu_rates in menus.items() for menu, rate in menu_rates.items()]\
                            , columns=['Category', 'Menu', 'Ratings'])
    memos_df = pd.DataFrame(list(memos.items()), columns=['Menu', 'Memo'])

    # menus, memos를 병합
    result_df = pd.merge(menus_df, memos_df, on='Menu')

    # 결과를 csv 파일로 저장
    result_df.to_csv('menus.csv', index=False)
    break

else:
    print("잘못된 번호를 선택했습니다. 다시 입력해 주세요.")
    continue
```

(4) Menu Recommendation - Select Category, Save Memo, Save Preferences

- Input; menus: a dictionary that stores categories,menus, and ratings.

memos: a dictionary that stores menus and memos.

Output: Menus and memos with newly changed and saved memos and prepernets for recommended menus

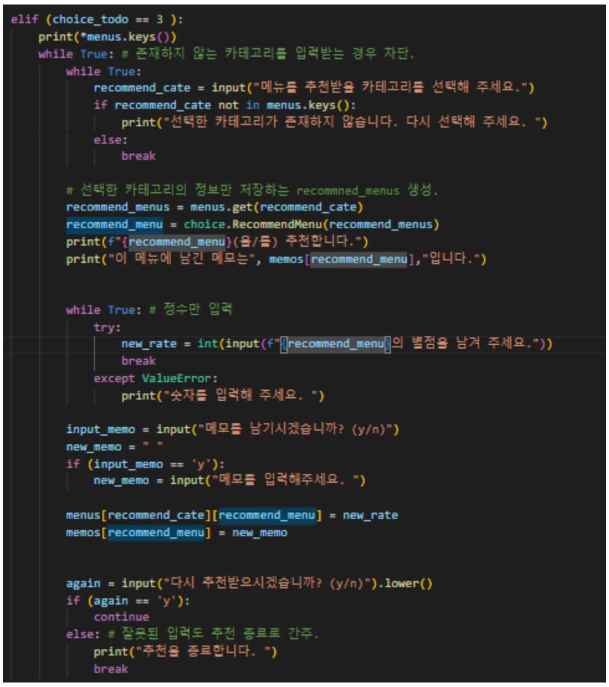
- Description: Enter the category for which the menu is recommended, and create a referral_menus that stores only the menus and ratings that correspond to that category in the menus.

In the Recommend Menu function, mix the order of referral_menus and loop each weight to select one random menu.

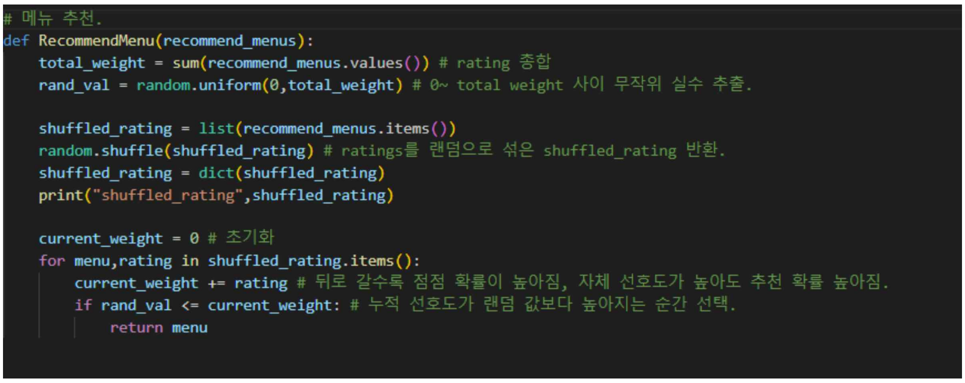
Specify a new rating on the selected menu and optionally leave a memo. You can be recommended again if you want.

- What you've learned: Repeat statement, conditional statement, function, module, exception

- Code Screenshot



```python
elif (choice_todo == 3 ):
    print(*menus.keys())
    while True: # 존재하지 않는 카테고리를 입력받는 경우 차단.
        while True:
            recommend_cate = input("메뉴를 추천받을 카테고리를 선택해 주세요.")
            if recommend_cate not in menus.keys():
                print("선택한 카테고리가 존재하지 않습니다. 다시 선택해 주세요. ")
            else:
                break

        # 선택한 카테고리의 정보만 저장하는 recommned_menus 생성.
        recommend_menus = menus.get(recommend_cate)
        recommend_menu = choice.RecommendMenu(recommend_menus)
        print(f"{recommend_menu}(을/를) 추천합니다.")
        print("이 메뉴에 남긴 메모는", memos[recommend_menu],"입니다.")


        while True: # 정수만 입력
            try:
                new_rate = int(input(f"{recommend_menu}의 별점을 남겨 주세요."))
                break
            except ValueError:
                print("숫자를 입력해 주세요. ")

        input_memo = input("메모를 남기시겠습니까? (y/n)")
        new_memo = " "
        if (input_memo == 'y'):
            new_memo = input("메모를 입력해주세요. ")

        menus[recommend_cate][recommend_menu] = new_rate
        memos[recommend_menu] = new_memo


        again = input("다시 추천받으시겠습니까? (y/n)").lower()
        if (again == 'y'):
            continue
        else: # 잘못된 입력도 추천 종료로 간주.
            print("추천을 종료합니다. ")
            break
```



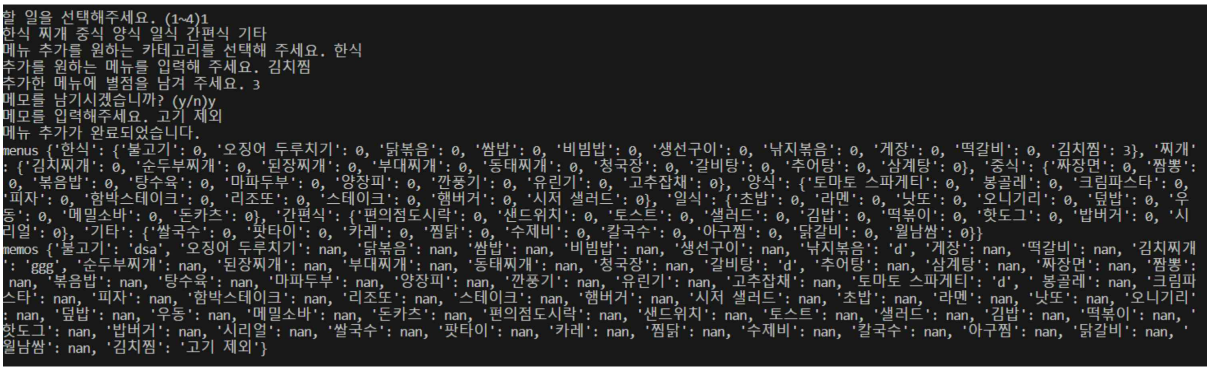
```python
# 메뉴 추천.
def RecommendMenu(recommend_menus):
    total_weight = sum(recommend_menus.values()) # rating 종합
    rand_val = random.uniform(0,total_weight) # 0~ total weight 사이 무작위 실수 추출.

    shuffled_rating = list(recommend_menus.items())
    random.shuffle(shuffled_rating) # ratings를 랜덤으로 섞은 shuffled_rating 반환.
    shuffled_rating = dict(shuffled_rating)
    print("shuffled_rating",shuffled_rating)

    current_weight = 0 # 초기화
    for menu,rating in shuffled_rating.items():
        current_weight += rating # 뒤로 갈수록 점점 확률이 높아짐, 자체 선호도가 높아도 추천 확률 높아짐.
        if rand_val <= current_weight: # 누적 선호도가 랜덤 값보다 높아지는 순간 선택.
            return menu
```

4. Test Result

(1) Add Menu

- Description: Test to see if the menu and ratings have been added as normal.

- Test Results Screenshot

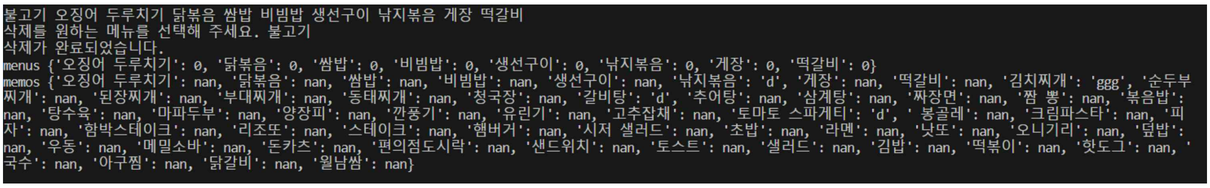


```
할 일을 선택해주세요. (1~4)1
한식 찌개 중식 양식 일식 간편식 기타
메뉴 추가를 원하는 카테고리를 선택해 주세요. 한식
추가를 원하는 메뉴를 입력해 주세요. 김치찜
추가한 메뉴에 별점을 남겨 주세요. 3
메모를 남기시겠습니까? (y/n)y
메모를 입력해주세요. 고기 제외
메뉴 추가가 완료되었습니다.
menus {'한식': {'불고기': 0, '오징어 두루치기': 0, '닭볶음': 0, '쌈밥': 0, '비빔밥': 0, '생선구이': 0, '낚지볶음': 0, '게장': 0, '떡갈비': 0, '김치찜': 3}, '찌개'
: {'김치찌개': 0, '순두부찌개': 0, '된장찌개': 0, '부대찌개': 0, '동태찌개': 0, '청국장': 0, '갈비탕': 0, '추어탕': 0, '삼계탕': 0}, '중식': {'짜장면': 0, '짬뽕':
 0, '볶음밥': 0, '탕수육': 0, '마파두부': 0, '양장피': 0, '깐풍기': 0, '유린기': 0, '고추잡채': 0}, '양식': {'토마토 스파게티': 0, ' 봉골레': 0, '크림파스타': 0,
'피자': 0, '함박스테이크': 0, '리조또': 0, '스테이크': 0, '햄버거': 0, '시저 샐러드': 0}, '일식': {'초밥': 0, '라멘': 0, '낫또': 0, '오니기리': 0, '덮밥': 0, '우
동': 0, '메밀소바': 0, '돈카츠': 0}, '간편식': {'편의점도시락': 0, '샌드위치': 0, '토스트': 0, '샐러드': 0, '김밥': 0, '떡볶이': 0, '핫도그': 0, '밥버거': 0, '시
리얼': 0}, '기타': {'쌀국수': 0, '팟타이': 0, '카레': 0, '찜닭': 0, '수제비': 0, '칼국수': 0, '아구찜': 0, '닭갈비': 0, '월남쌈': 0}}
memos {'불고기': 'dsa', '오징어 두루치기': nan, '닭볶음': nan, '쌈밥': nan, '비빔밥': nan, '생선구이': nan, '낚지볶음': 'd', '게장': nan, '떡갈비': nan, '김치찌개
': 'ggg', '순두부찌개': nan, '된장찌개': nan, '부대찌개': nan, '동태찌개': nan, '청국장': nan, '갈비탕': 'd', '추어탕': nan, '삼계탕': nan, '짜장면': nan, '짬뽕':
 nan, '볶음밥': nan, '탕수육': nan, '마파두부': nan, '양장피': nan, '깐풍기': nan, '유린기': nan, '고추잡채': nan, '토마토 스파게티': 'd', ' 봉골레': nan, '크림파
스타': nan, '피자': nan, '함박스테이크': nan, '리조또': nan, '스테이크': nan, '햄버거': nan, '시저 샐러드': nan, '초밥': nan, '라멘': nan, '낫또': nan, '오니기리'
: nan, '덮밥': nan, '우동': nan, '메밀소바': nan, '돈카츠': nan, '편의점도시락': nan, '샌드위치': nan, '토스트': nan, '샐러드': nan, '김밥': nan, '떡볶이': nan, '
핫도그': nan, '밥버거': nan, '시리얼': nan, '쌀국수': nan, '팟타이': nan, '카레': nan, '찜닭': nan, '수제비': nan, '칼국수': nan, '아구찜': nan, '닭갈비': nan, '
월남쌈': nan, '김치찜': '고기 제외'}
```

(2) Delete Menu

- Description: Test to see if the menu and rating have been successfully deleted.

- Test Results Screenshot

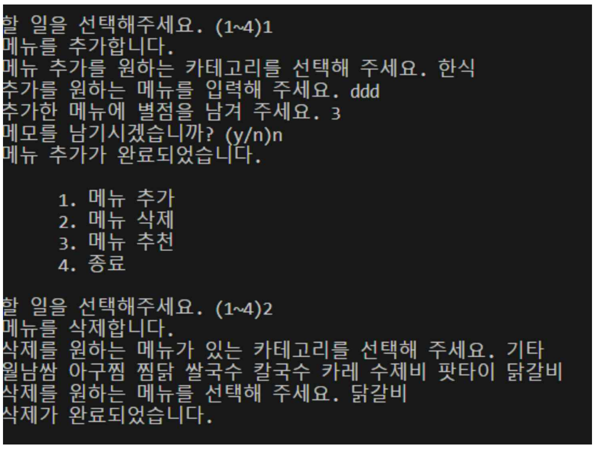


```
불고기 오징어 두루치기 닭볶음 쌈밥 비빔밥 생선구이 낚지볶음 게장 떡갈비
삭제를 원하는 메뉴를 선택해 주세요. 불고기
삭제가 완료되었습니다.
menus {'오징어 두루치기': 0, '닭볶음': 0, '쌈밥': 0, '비빔밥': 0, '생선구이': 0, '낚지볶음': 0, '게장': 0, '떡갈비': 0}
memos {'오징어 두루치기': nan, '닭볶음': nan, '쌈밥': nan, '비빔밥': nan, '생선구이': nan, '낚지볶음': 'd', '게장': nan, '떡갈비': nan, '김치찌개': 'ggg', '순두부
찌개': nan, '된장찌개': nan, '부대찌개': nan, '동태찌개': nan, '청국장': nan, '갈비탕': 'd', '추어탕': nan, '삼계탕': nan, '짜장면': nan, '짬 뽕': nan, '볶음밥':
nan, '탕수육': nan, '마파두부': nan, '양장피': nan, '깐풍기': nan, '유린기': nan, '고추잡채': nan, '토마토 스파게티': 'd', ' 봉골레': nan, '크림파스타': nan, '피
자': nan, '함박스테이크': nan, '리조또': nan, '스테이크': nan, '햄버거': nan, '시저 샐러드': nan, '초밥': nan, '라멘': nan, '낫또': nan, '오니기리': nan, '덮밥':
nan, '우동': nan, '메밀소바': nan, '돈카츠': nan, '편의점도시락': nan, '샌드위치': nan, '토스트': nan, '샐러드': nan, '김밥': nan, '떡볶이': nan, '핫도그': nan, '
국수': nan, '아구찜': nan, '닭갈비': nan, '월남쌈': nan}
```

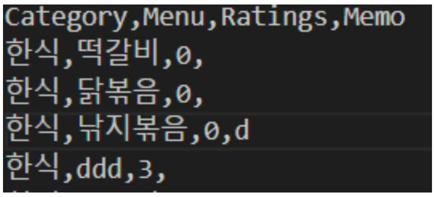
(3) File Storage Function

- Description: Test to ensure that the changes are stored in the file properly.

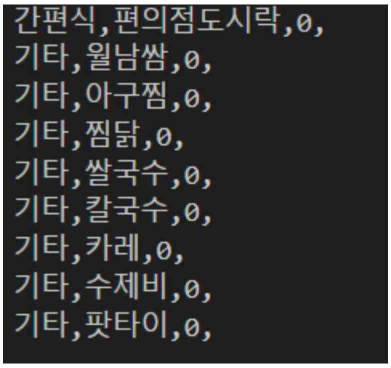
- Test Results Screenshot



```
할 일을 선택해주세요. (1~4)1
메뉴를 추가합니다.
메뉴 추가를 원하는 카테고리를 선택해 주세요. 한식
추가를 원하는 메뉴를 입력해 주세요. ddd
추가한 메뉴에 별점을 남겨 주세요. 3
메모를 남기시겠습니까? (y/n)n
메뉴 추가가 완료되었습니다.

     1. 메뉴 추가
     2. 메뉴 삭제
     3. 메뉴 추천
     4. 종료

할 일을 선택해주세요. (1~4)2
메뉴를 삭제합니다.
삭제를 원하는 메뉴가 있는 카테고리를 선택해 주세요. 기타
월남쌈 아구찜 찜닭 쌀국수 칼국수 카레 수제비 팟타이 닭갈비
삭제를 원하는 메뉴를 선택해 주세요. 닭갈비
삭제가 완료되었습니다.
```



```
Category,Menu,Ratings,Memo
한식,떡갈비,0,
한식,닭볶음,0,
한식,낚지볶음,0,d
한식,ddd,3,
```

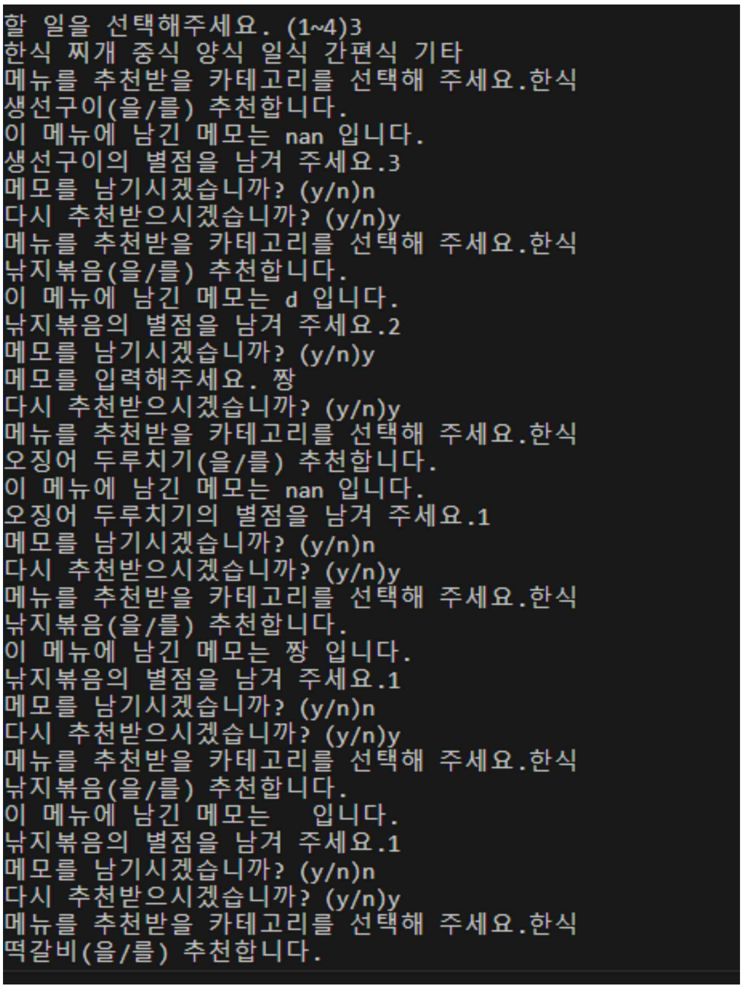


```
간편식,편의점도시락,0,
기타,월남쌈,0,
기타,아구찜,0,
기타,찜닭,0,
기타,쌀국수,0,
기타,칼국수,0,
기타,카레,0,
기타,수제비,0,
기타,팟타이,0,
```

(4) Menu Recommendation

- Description: Check that menu recommendations are being made randomly, and test to ensure that information is entered correctly.

- Test Results Screenshot



```
할 일을 선택해주세요. (1~4)3
한식 찌개 중식 양식 일식 간편식 기타
메뉴를 추천받을 카테고리를 선택해 주세요.한식
생선구이(을/를) 추천합니다.
이 메뉴에 남긴 메모는 nan 입니다.
생선구이의 별점을 남겨 주세요.3
메모를 남기시겠습니까? (y/n)n
다시 추천받으시겠습니까? (y/n)y
메뉴를 추천받을 카테고리를 선택해 주세요.한식
낚지볶음(을/를) 추천합니다.
이 메뉴에 남긴 메모는 d 입니다.
낚지볶음의 별점을 남겨 주세요.2
메모를 남기시겠습니까? (y/n)y
메모를 입력해주세요. 짱
다시 추천받으시겠습니까? (y/n)y
메뉴를 추천받을 카테고리를 선택해 주세요.한식
오징어 두루치기(을/를) 추천합니다.
이 메뉴에 남긴 메모는 nan 입니다.
오징어 두루치기의 별점을 남겨 주세요.1
메모를 남기시겠습니까? (y/n)n
다시 추천받으시겠습니까? (y/n)y
메뉴를 추천받을 카테고리를 선택해 주세요.한식
낚지볶음(을/를) 추천합니다.
이 메뉴에 남긴 메모는 짱 입니다.
낚지볶음의 별점을 남겨 주세요.1
메모를 남기시겠습니까? (y/n)n
다시 추천받으시겠습니까? (y/n)y
메뉴를 추천받을 카테고리를 선택해 주세요.한식
낚지볶음(을/를) 추천합니다.
이 메뉴에 남긴 메모는   입니다.
낚지볶음의 별점을 남겨 주세요.1
메모를 남기시겠습니까? (y/n)n
다시 추천받으시겠습니까? (y/n)y
메뉴를 추천받을 카테고리를 선택해 주세요.한식
떡갈비(을/를) 추천합니다.
```

5. Changes in Comparison to the Plan

No changes

6. Lessons Learned & Feedback

I left a lot of regrets about what I did wrong about the project, but I think it was a good experience and a good class.