Dog Shelter Feeding Helper

T1A3 Terminal Application by James Boland
for CoderAcademy.

What is it?

Dog Shelter Feeding Helper is an application to manage a dog shelter's daily feeding schedule, keeping track of the dogs that are in the shelter, and if they have or have not been fed for the day.

The 'fed' status of the dog will reset each day.

Dogs information will be stored in a database (JSON) where it will remain until it has been deleted (adopted).

The notion of it being a dog shelter can easily be changed with some minor adjustments to the code. Another two examples of how this application could be used would be to track meals at a boarding school, or medication management at a nursing home.

The features

The main features of the app include:

1. Continuity message to indicate to user if fed count has been reset (checks if new day or same day as last session)
2. Display of count of dogs in shelter, and count of dogs that have been fed today.
   a. Resets each day
   b. Uses information from JSON file. Changes in other features affect this count
3. Main menu, including:
   a. View dogs in shelter
   b. Add a new dog to shelter (name, breed, medical & dietary requirements, details of medical & dietary requirements, fed status)
   c. Edit dog information (same as above)
   d. Update the fed status of a dog, marking them as fed
   e. View dogs in shelter still to be fed
   f. Remove a dog from the shelter (and database)
   g. Exit

The main ingredients

Fed count tally:

      dogsdb.json

      date.txt

Logic:

      Functions

      Variables

      If/else statements

      Try/except error handling

      While & for loops

      Using data from date.txt & dogsdb.json

Testing

Test 1: Fed dog count tally

- Testing all the ways that the program *should* affect the fed dog count tally.

Test 2: Main menu functionality

- Testing that the most used feature of the application, the main menu, runs as intended and will not crash due to user error.

Feature being tested	Test case	Test data	Expected result	Actual Result	Status
Fed dog count tally	Check results by adding a new dog with the answer "Yes" to the prompt 'Has (dog) been fed today? in the new dog menu	In new dog menu (name, breed, MDR, DMDR, fed): name: 'test' breed: 'test breed' mdr: 'no' dmdr: n/a fed: 'Yes'	A new dog is created with its fed status as 'Yes'. Tally on main menu is updated and entry of new dog in json file has 'Fed' key value set to 'Yes'	As intended	Pass
	Check results by updating a dogs fed status in the 'update fed status' menu	User enters a valid existing DoggyID and updates a dogs fed status from 'No' to 'Yes'	User receives confirmation of dogs status being updated, tally on main menu is updated, dog matching DoggyID has 'Fed' key value set to 'Yes' in json file.	As intended	Pass
	Check results by removing a dog from 'delete dog from database' menu	User enters a valid existing DoggyID and confirms removal of a dog from database	User receives confirmation of dog being removed, dog is removed from dogsdb.json completely, total dogs in tally drops by 1, if dog that was removed was fed, fed dog tally is dropped by 1, if dog that was removed wasn't fed, fed dog tally remains same.	As intended	Pass
	Check results by changing 'fed' status to 'yes' in edit dog menu	In edit dog menu, user enters valid existing DoggyID and updates a dogs fed status from 'No' to 'Yes'	User receives confirmation of dogs status being updated, tally on main menu is updated, dog matching DoggyID has 'Fed' key value set to 'Yes' in json file.	As intended	Pass
	Check results if tally is reset on new day (real)	Session is started on new day, previous day had dogs in database with 'Fed' key value of 'Yes'. Tested by actually using program on different day	Program checks date.txt, recognises new day, writes current date. Function update_to_no() is triggered; dogs in db have 'Fed' key value updated to 'No' and updates fed dog counter on main menu. User receives message notifying that the fed count has been reset	As intended	Pass
	Check results if tally is reset on new day (simulated)	Session is started on same day, previous day had dogs in database with 'Fed' key value of 'Yes'. Tested by changing date in 'date.txt'	Program checks date.txt, recognises new day, writes current date. Function update_to_no() is triggered; dogs in db have 'Fed' key value updated to 'No' and updates fed dog counter on main menu. User receives message notifying that the fed count has been reset	As intended	Pass
	Check results if tally is recovered from same day	Session is started on same day, previous day had dogs in database with mixed 'Fed' key values of 'Yes' and 'No'.	Program checks date.txt, recognises same day, recovers db from last session without making any changes to db. Dogs fed status and total fed counter does not change from last session. User receives message notifying fed count has been recovered from last session	As intended	Pass

Test case	Test data	Expected result	Actual Result	Status
User inputs valid menu option (1-7)	User inputs 1, 2, 3, 4, 5, 6 or 7	Takes user to expected page for each entry	As intended	Pass
	User inputs 8, 9, 0, 11, 27 etc.	Triggers else statement, informing the user of an invalid input and allowing user to return to main menu	As intended	Pass
	User inputs a string	Triggers ValueError and prints error message, giving user option to return to main menu.	Triggers the else statement, informing the user of an invalid input and allows user to return to main menu	F - minor severity as actual result produces same outcome for user.
	User hits 'enter' without entering any input	Triggers the else statement, informing the user of an invalid input and allows user to return to main menu	As intended	Pass
	User enters a character ($, ^, %, [, &)	Triggers ValueError and prints error message, giving user option to return to main menu.	Triggers the else statement, informing the user of an invalid input and allows user to return to main menu	F - minor severity as actual result produces same outcome for user.