Terminal App Slide Deck

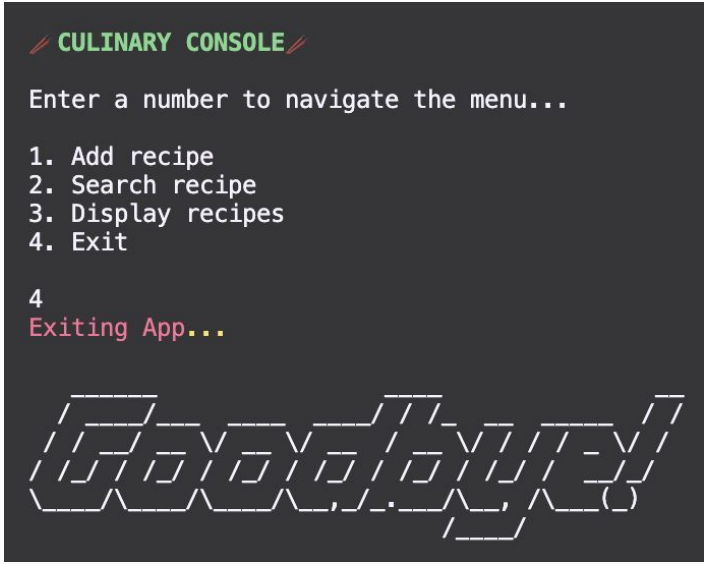
Elijah Fernandez

Terminal App Walkthrough

Culinary Console is a Terminal based Recipe Application.

Main Features:

- Add recipe
- Search for a recipe using keywords
- Display all recipes
- Exit app
- Data saved in JSON file



Logic

```
def load_recipe_list():
    try:
        with open(json_file, "r") as infile:
            recipe_list = json.load(infile)
    except FileNotFoundError:
        print("The 'recipes.json' file is not found...")
        print("Creating a new recipe book...")
        recipe_list = []

    return recipe_list
```

Demo

Programing concepts

Implement features in the software development plan you have designed. You must utilise a range of programming concepts and structures using Python such as:

- variables and variable scope

- loops and conditional control structures

- write and utilise simple functions

- error handling

- input and output

- importing a Python package

- using functions from a Python package

Tests

Design TWO tests which check that the application is running as expected.

Each test should:

- cover a different feature of the application

- state what is being tested

- provide at least TWO test cases and the expected results for each test case

> An **outline** of the testing procedure and cases should be included with the source code of the application

Tests

TEST1: test_file_creation:

Ensure that a file is created when the create_recipe_file function is called.

The test checks if the file doesn't exist before calling the function, and then verifies that the file exists after the function is executed.

TEST2: test_add_recipe:

Test the process of adding a recipe to the recipe list and saving it to a file.

The test creates a file, loads the recipe list from the file, adds a sample recipe to the list, and saves the updated list back to the file.

It then checks if the added recipe is present in the loaded list by comparing various attributes of the recipe.

Test 1: File Creation

Feature Covered

The first test verifies the file creation functionality of the application.

What is Being Tested

This test checks whether the create_recipe_file function successfully creates a file when called.

Test Cases and Expected Results

Test Case: File doesn't exist before the function call

Expected Result: The file should be created after calling the create_recipe_file function.

Test Case: File already exists before the function call

Expected Result: The file should still be created and overwritten with an empty file.

Test 2: Adding a Recipe

Feature Covered

The second test validates the process of adding a recipe to the recipe list and saving it to a file.

What is Being Tested

This test ensures that the application correctly handles adding a recipe, updating the recipe list, and saving it to a file.

Test Cases and Expected Results

Test Case: Adding a recipe to an empty list

Expected Result: The recipe should be successfully added to the list, and the updated list should be saved to the file.

Test Case: Adding a recipe to a non-empty list

Expected Result: The recipe should be appended to the existing list, and the updated list should be saved to the file.

Help Documentation

You must include:

- steps to install the application

- any dependencies required by the application to operate

- any system/hardware requirements

- how to use any command line arguments made for the application

Help Documentation

Steps to install:

1. download zip file from: https://github.com/elijahjf/elijahfernandez_T1A3
2. unzip
3. in terminal navigate to "elijahfernandez_T1A3\src"
4. enter "ls -a" into the terminal to show all files including hidden ones in the src folder
5. run the file script.sh
6. if you're having a permissions error, type in "chmod -x script.sh"
7. do the same to run tests

Dependencies required by the application to operate

See requirements.txt

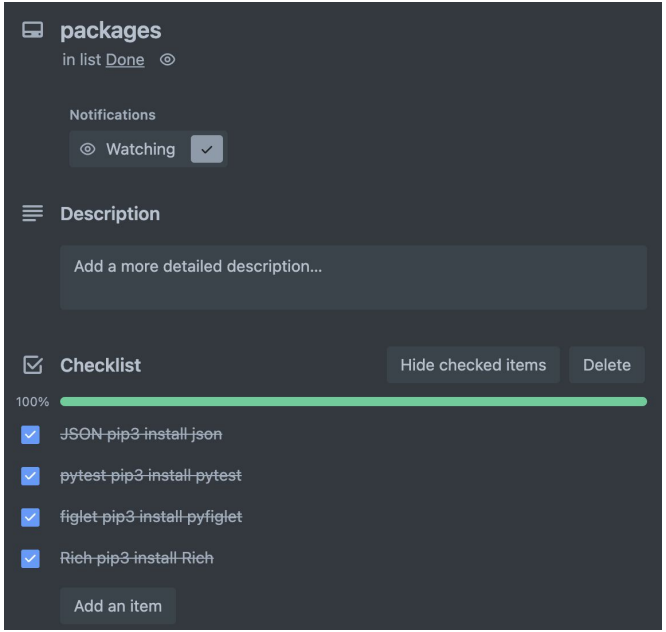
System/hardware requirements

- Ensure you have Python 3 and pip3 installed on your computer.
- Using a UNIX supporting terminal
- computer's administor permissions

The Development and Build Process

App Development

I found this to be a challenging but rewarding process. I was behind on a previous project which gave me less time than I'd like to work on this. It proved to me how important it is to time manage which is an important part of this project. Using trello and git really helped me track my progress. I honestly wasn't sure if I was ready to actually make an app - but I can see now that the only way to do it is to try. I went back on a lot of previous learning material which was quite helpful, especially for testing and utilising packages.



Challenges

I moved my main.py file into an src folder and then my json file stopped working correctly because it was no longer in the same directory as the relevant scripts. It took a lot of trail and research to figure out how to fix a seemingly simple problem, because of code that was made when the files were created that I can't see usually. It did help me learn more about the process deeper than I would have otherwise if I didn't encounter this issue.

Ethical Issues

Intellectual Property: Recipes are personal to a lot of people. It is important to respect the intellectual property of others. As well as avoid infringing on copyrights or plagiarism by giving credit to recipe authors and/or obtaining permission to put them on the app. This is why I only have displayed original or made up recipes. I could also use public domain recipes, or properly licensed ones if I wanted to include a good catalogue for the user to begin with.

Accessibility: It is important that apps are accessible as possible. I tried to use colors that are easy to read on the background. I also tried to use simple words that are easy to understand.

Favorite Parts

To be honest, my favourite part is that it works. Just having an application that works is something I am really proud of.

Thank you

References:

-