CAREERFOUNDRY

Achievement 1 Project Brief: Recipe App (Command Line Version)

Objective

Build the command line version of a Recipe app, which acts as a precursor to its web app counterpart in Achievement 2.

Context

Building a website with the Django web framework is almost entirely done in Python, and takes advantage of Python's object-oriented nature. You may also be required to modify your code in the event of any updates to Django—here, it helps to understand the instructions and syntaxes provided. Debugging for errors and exceptions also becomes an easy task with the concepts outlined in this Achievement.

This project primarily focuses on learning Python fundamentals, data structures, and object-oriented programming. You'll also learn how to interact with databases using Python, which will help you when you have to do the same with the Django framework. The project also aims to teach you standard programming practices that will make your code simpler, easier to read, and robust during execution.

The 5 W's

1. **Who**—Your professional network and potential employers. Potential employers can view this project as a demonstration of your skills in Python. Achievement 2 is where you'll deliver a project worthy of portfolio inclusion, while Achievement 1 can illustrate your Python scripting skills.
2. **What**—A command line application for a Recipe app, which is able to create, read, and modify recipes, as well as searching for recipes based on ingredients.
3. **When**—The code will be readily available on your GitHub repository, and will require a local installation of Python 3.6+ to run it on any machine, which is fairly simple to set up. Recruiters can access this code easily to view how you've implemented your project.
4. **Where**—Your code can be run on any machine that has an active Python installation. You may either use your GitHub profile or your portfolio site to host the code for your project for potential employers and your professional network to view.
5. **Why**—This Achievement is geared towards exhibiting to your professional network and potential employers 1) your understanding of the fundamentals of Python programming, and 2) the potential to apply this knowledge when working with web application frameworks like Django.

User Goals

Your users should be able to create and modify recipes with ingredients, cooking time, and a difficulty parameter that would be automatically calculated by the app. Your users should also be able to search for recipes by their ingredients.

Key Features

- Create and manage the user's recipes on a locally hosted MySQL database.
- Option to search for recipes that contain a set of ingredients specified by the user.
- Automatically rate each recipe by their difficulty level.
- Display more details on each recipe if the user prompts it, such as the ingredients, cooking time, and difficulty of the recipe.

Technical Requirements

- The app should handle any common exceptions or errors that may pop up either during user input, database access, for example, and display user-friendly error messages.
- The app must connect to a MySQL database hosted locally on your system.
- The app must provide an easy-to-use interface, supported by simple forms of input and concise instructions that any user can follow—always assume that they aren't as technically proficient as you may be. For instance, if the program requires that the user picks an option from a list, instead of having them manually type in the option, list the options with numbers, and have them enter the number corresponding to their choice.
- The app should work on Python 3.6+ installations.
- App code must be well-formatted according to standardized guidelines
- App code should also be supported by concise, helpful comments that illustrate the flow of the program.

Nice to Have

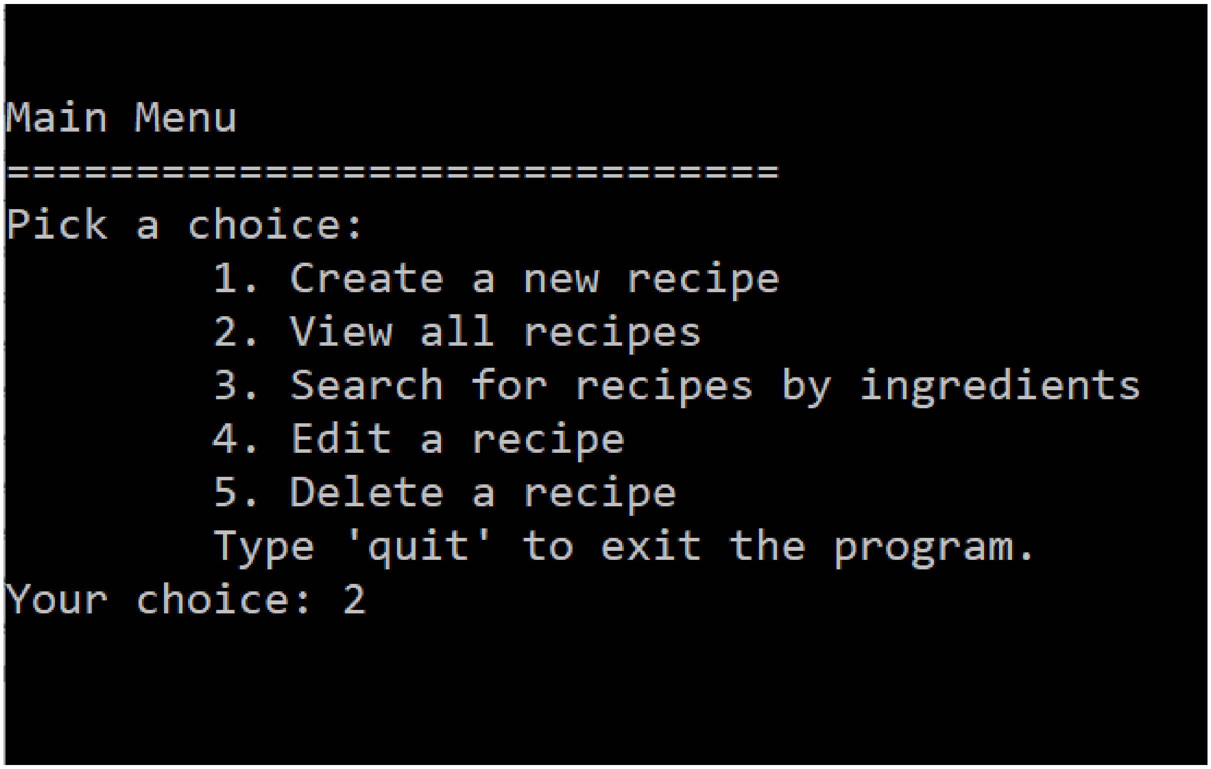
- Ideally, the application should be able to handle any kind of input from the user— assume that your user would enter random, nonsensical inputs (this concept is known as "monkey testing").

Although monkey testing is formally an automated process, you can try this yourself manually with your own set of inputs. Test the limits of your application by entering erroneous choices, random characters, or long strings.

- Make an instruction manual! A simple, one-page document would do: it should describe the features of your application, as well as simple instructions that can take the user through your program. You can use the README file of your GitHub repository for hosting this instruction manual.

Mockups or Other Assets

All input and output for this app takes place from a command line, so there's a limited amount you can do in terms of presentation. Although you're free to communicate in your own words for the interface, here is a general template that you can go through to build an approximate idea of your project. Once you're comfortable coding in Python, we encourage you to delve further into making your app user-friendly. Your command line “Main Menu” should look something like this:



```
Main Menu
==============================
Pick a choice:
        1. Create a new recipe
        2. View all recipes
        3. Search for recipes by ingredients
        4. Edit a recipe
        5. Delete a recipe
        Type 'quit' to exit the program.
Your choice: 2
```

Your Project Deliverables

Throughout this Achievement, you'll be working from one Exercise to the next to build on different aspects of your app. In the task at the end of each Exercise, you'll submit a deliverable that adds one or more features to your app, each contributing to your final product—in this case, a Recipe app that stores and searches for recipes.

Here is a breakdown of your course project deliverables by Exercise:

Exercise 1: Intro to Python Programming

- Install Python on macOS, Windows, or Linux
- Create and manage virtual environments
- Use pip to install and manage packages

Exercise 2: Data Types in Python

- Use data types and methods to execute Python commands that store recipes containing their own internal data
- Enter a number of these recipes into another linear data structure

Exercise 3: Functions and Other Operations in Python

- Create your first script on a .py script file
- Build a script that uses *if-elif-else* statements, *for* loops, and functions to take recipes from the user then display them

Exercise 4: File Handling in Python

- Create a Python script that takes recipes from the user and writes the data in a binary file
- Create another script that reads the binary file and lists out the available ingredients. The user chooses an ingredient and the script displays all recipes which contain it
- Use Python's exception handling features to handle common errors

Exercise 5: Object-Oriented Programming in Python

- Build a custom class for your recipes, which contains its own data attributes for name, ingredients, cooking time, and difficulty, as well as other custom methods to interact with this data

Exercise 6: Connecting to Databases in Python

- Set up a MySQL database and connect your scripts to it
- Build an application that creates, reads, updates, and deletes recipes, as well as searching for them by a single ingredient

Exercise 7: Finalizing Your Python Program

- Use an Object Relational Mapper from SQLAlchemy to manage the contents of your database from your application
- Build a user-friendly menu for entering and searching recipes and ingredients
- Store recipe and ingredient data in a MySQL database
- Implement recipe search according to user-defined ingredients
- Implement a detailed display of the recipe selected by the user

Optional: Advanced Deliverables

You can add as many features to your Recipe app as you like. The following two approaches, for example, can make your app more engaging for users:

- Instead of the user selecting ingredients from a list for their recipe search, you could implement a search entry where the program searches for ingredients closest to the user's search terms. For example, if a recipe contains "carrots", but the user searches for "carrot", your application would still bring up the relevant recipes
- You could also add more attributes to the Recipe class for expanded app functionality. For example: recipe instructions, or quantities for each ingredient