Exercise 1.7: Object-Relational Mapping in Python

Reflection Questions :

- *What is an Object Relational Mapper and what are the advantages of using one?*
    - **A:** ORM's (Object-Relational Mappers) are tools that make interacting with DBMS's (DataBase Management Systems) much easier. It does this by translating the contents of the database into classes and objects that are more easily directly interactable with Python. This can cut out countless potential syntax errors as well as a lot of developer tedium.

- *By this point, you've finished creating your Recipe app. How did it go? What's something in the app that you did well with? If you were to start over, what's something about your app that you would change or improve?*
    - **A:** The 'edit_recipe()' function is where I was most satisfied with my efforts. There were several steps in which many things could go wrong, as well as a potentially problematic relationship with the 'calculate_difficulty()' function, and I was able to achieve the basic functionality with relative efficiency.

      The 'edit_recipe()' function is also where I intend to return to this project to make improvements that weren't part of the Exercise: I want to make the steps of the recipe-editing process to be more precise, as well as to update the ingredients by adding or removing ingredients, as opposed to its current behavior where it simply has the user re-enter the entire ingredients list.

- *Imagine you're at a job interview. You're asked what experience you have creating an app using Python. Taking your work for this Achievement as an example, draft how you would respond to this question.*
    - **A:** I have experience with Python that includes developing an app allowing users to easily manage their own data. I configured a database using MySQL and SQL Alchemy that can be used by any user to add, remove, and edit

entries that can be accessed at any time, as well as re-visited and found by way of searching the items within each data entry. In this case, I used cooking recipes. The user could store recipes, the corresponding ingredients, and the cooking time. The user could return to their saved recipes and pull them up from a list, or search recipes by the ingredients within. Any of these recipe entries could be edited at any time, or removed at the user's discretion. These recipes were also analyzed by a function I created that determined each recipes difficulty level. While this particular app used cooking recipes, I am confident I can create any number of countless similar apps for data entry, with user-friendly prompts for organizing all of the corresponding information within each entry, as well as adding functions for analyzing the data, and organizing and filtering returned outputs according any logical criteria. As well, this experience has equipped me to be ready to manage databases of this kind in Python environments.

- *You've finished Achievement 1! Before moving on to Achievement 2, take a moment to reflect on your learning in the course so far:*

  - *What went well during this Achievement?*
    - A: This achievement was almost entirely focused on the coding logic, as opposed to making many things work across a plethora of internet locations and other dependent programs, packages, libraries, etc. This was something I needed, as I was previously getting less programmatic logic work than I desired.

  - *What's something you're proud of?*
    - A: I'm proud that I have had all of my Task submissions approved on the first try, so far. While this did take a substantial amount more effort in being attentive to every detail, it was worth it, in overall time efficiency as well as a much needed sense of accomplishment. In this achievement, unlike past ones, when I got stuck, I still always had a sense of things I could try and research to keep working towards the solutions.

  - *What was the most challenging aspect of this Achievement?*
    - A: Unlike past achievements, there was much less hand-holding. This one was more traditional, in that, the curriculum taught the concepts, and then the task was separate and required the formation of the code on my own. While the **tasks** had much more description of what to do, they weren't step-by-step walkthroughs, as they were in past achievements' reading material.

- *Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?*
  - A: It did. I have more confidence in the Python work than I do in the work from previous Achievements. While the past achievements resulted in much more glamorous and exciting end products, I always felt very lost in what I was supposed to be doing. That being said, JavaScript is most likely where I will be finding a career path.

- *What's something you want to keep in mind to help you do your best in Achievement 2?*
  - A: While my time (in my very busy life) is precious, I can actually **save** time by just walking away in some circumstances. Quite often, in this coding work, there are some elusive yet simple problems that are easy to see when I return with a fresh mind. In kind, when I attack one single mystery for too long in one sitting, I often start going down the wrong paths, and even creating new problems for myself.