**Week 3 creation expense tracker**

code appears to be an Expense Tracker program written in Python. It allows users to add expenses, view a list of all expenses, and view a summary of monthly expenses by category. The data is stored in a JSON file, and the program reads and writes to this file to maintain the user's expense data.

Here's a brief overview of the key components:

**ExpenseTracker Class:**

* + **Constructor (\_\_init\_\_):** Initializes the expense tracker with an empty list of expenses and an empty set of categories. It then calls **load\_data** to load any existing data from the 'expenses.json' file.
  + **load\_data method:** Loads expense data from 'expenses.json' if the file exists.
  + **save\_data method:** Saves the current state of expenses and categories to 'expenses.json'.
  + **add\_expense method:** Adds a new expense to the tracker with the provided amount, description, and category. It then saves the updated data using **save\_data**.
  + **view\_expenses method:** Prints a list of all expenses, including the date, amount, description, and category.
  + **view\_summary method:** Prints a summary of monthly expenses by category.
  + **run method:** Provides a simple command-line interface for users to interact with the expense tracker. It includes options to add expenses, view all expenses, view a summary, and exit the program.

**Main Block (`if name == "main":):**

* + Creates an instance of the **ExpenseTracker** class.
  + Invokes the **run** method to start the program.

**Data Storage:**

Expense data is stored in a JSON file ('expenses.json').

Data is loaded from this file when the program starts, and it is saved back to the file after each modification (adding a new expense).

**The** **User Interface:**

program presents a simple command-line menu with options to add expenses, view all expenses, view a monthly summary, or exit the program.

User input is used to determine the action to be taken.