Personal Expense Tracker - Full Explanation

1. Importing Modules

- `csv`: Python’s built-in library to work with CSV files (rows/columns).

- `datetime`: Used to work with dates and times.

2. Global Variable

- `FILENAME = "expenses.csv"` → stores all expenses in a CSV file.

3. Saving Expenses

- `save_expense(date, category, amount, note="")`

- Opens CSV file in append mode (`'a'`).

- Uses `csv.writer` to write one row (date, category, amount, note).

- Prints confirmation.

4. Reading Expenses

- `read_expenses()`

- Opens CSV file in read mode (`'r'`).

- Uses `csv.reader` to read all rows as a list.

- Returns empty list if file not found (first run).

5. Printing Expenses

- `print_expenses(expenses)`

- Prints data in a neat tabular format using f-strings.

- Aligns columns with `:11` or `:8` spacing.

- Handles empty list gracefully.

6. Filtering Expenses

- `filter_expenses()` provides filtering options:

1. Specific Date → matches date exactly (`YYYY-MM-DD`).

2. Date Range → converts strings to `datetime` objects and checks if expense falls between start and end date.

3. Category → matches case-insensitive category.

4. Show all → prints all records.

7. Main Program

- `main()` shows a looped menu:

1. Add Expense

2. View Expenses (with filters)

3. Exit

- While loop keeps program running until user exits.

- Input validation ensures correct formats:

- Date defaults to today if blank.

- Category required (cannot be empty).

- Amount must be a number (`float` conversion).

- Note is optional.

8. Program Execution

- `if __name__ == "__main__": main()` ensures the program runs only when executed directly, not when imported.

Summary:

This project is a **CLI Expense Tracker** that:

- Stores expenses in a CSV file.

- Lets users add, view, and filter expenses.

- Prints expenses neatly in table format.

- Uses Python built-in libraries only (lightweight and portable).