Personal Expense Tracker — Full Project Documentation

1. Overview This is a command-line Personal Expense Tracker built with Python. It allows the user to add, view, and filter expenses while saving all records into a CSV file. The program only uses Python's standard library, making it lightweight and portable. 2. Core Features Add new expenses with date, category, amount, and optional note. Save expenses persistently into a CSV file. Read and display expenses in a tabular format. Filter expenses by specific date, date range, or category. User-friendly input validation and error handling. 3. Control Flow The program starts in main() and shows a looped menu. Option 1 lets the user add a new expense and saves it to the CSV file. Option 2 opens the filtering menu to view specific expenses. Option 3 exits the program. 4. Functions **save_expense**: Saves a new row to the CSV file. **read_expenses**: Reads and returns all rows from the CSV file. **print_expenses**: Prints expenses in a neat table format. **filter_expenses**: Lets the user filter records by date, range, or category. **main**: The main loop handling the menu and user interactions. 5. Possible Improvements Add totals and category-wise summaries. Include a CSV header for better structure. Sort results by date when printing. Switch to a database (e.g., SQLite) for more advanced usage.

Fully Annotated Source Code

```
import csv  # Provides functions to read/write CSV files (for storing expenses)
from datetime import datetime  # Used for working with and validating dates

# The name of the CSV file where expenses will be stored
FILENAME = "expenses.csv"

def save_expense(date, category, amount, note=""):
    """
    Save a single expense to the CSV file.
    - date: str, format 'YYYY-MM-DD'
    - category: str, type of expense (Food, Travel, etc.)
    - amount: float, how much was spent
    - note: str, optional description
    """
    with open(FILENAME, mode='a', newline='') as file:  # Open file in append mode
        writer = csv.writer(file)  # Create a CSV writer
        writer.writerow([date, category, amount, note])  # Save expense row
    print("Expense added successfully!")  # Feedback to user

def read_expenses():
    """
    Read all expenses from the CSV file.
    Returns a list of rows.
    """
    try:
        with open(FILENAME, mode='r', newline='') as file:  # Open for reading
            reader = csv.reader(file)  # CSV reader object
            return list(reader)  # Return rows as a list
    except FileNotFoundError:
        return []  # Return empty if no file exists yet

def print_expenses(expenses):
    """
    Print a list of expenses in a formatted table.
    """
    if not expenses:  # If no expenses, inform user
        print("No expenses found.")
        return

    # Print header
    print("\nDate        | Category    | Amount   | Note")
    print("-----------------------------------------------")

    # Loop through each expense and print neatly
    for row in expenses:
        date, category, amount, note = row
        print(f"{date:11} | {category:11} | {amount:8} | {note}")

def filter_expenses():
    """
    Allow user to filter expenses by:
    - Specific date
    - Date range
    - Category
    - Or view all
    """
    expenses = read_expenses()  # Load expenses

    if not expenses:  # If no data
        print("No expenses recorded yet.")
        return

    # Show menu
    print("\nFilter by:")
    print("1. Specific Date")
    print("2. Date Range")
    print("3. Category")
    print("4. Show all")
    choice = input("Enter choice (1/2/3/4): ").strip()
```

```
    if choice == '1':
        date_input = input("Enter date (YYYY-MM-DD) or leave blank: ").strip()
        if not date_input:
            print("Filter cancelled.")
            return
        filtered = [e for e in expenses if e[0] == date_input]
        print_expenses(filtered)

    elif choice == '2':
        start_date = input("Enter start date (YYYY-MM-DD): ").strip()
        end_date = input("Enter end date (YYYY-MM-DD): ").strip()
        try:
            start_dt = datetime.strptime(start_date, "%Y-%m-%d")
            end_dt = datetime.strptime(end_date, "%Y-%m-%d")
        except ValueError:
            print("Invalid date format. Use YYYY-MM-DD.")
            return

        if start_dt > end_dt:
            print("Start date must be before end date.")
            return

        filtered = []
        for e in expenses:
            try:
                exp_date = datetime.strptime(e[0], "%Y-%m-%d")
                if start_dt <= exp_date <= end_dt:
                    filtered.append(e)
            except ValueError:
                continue
        print_expenses(filtered)

    elif choice == '3':
        category_input = input("Enter category: ").strip().lower()
        if not category_input:
            print("Filter cancelled.")
            return
        filtered = [e for e in expenses if e[1].lower() == category_input]
        print_expenses(filtered)

    elif choice == '4':
        print_expenses(expenses)
    else:
        print("Invalid choice.")

def main():
    """
    Main loop of the program.
    Lets user add expenses, view expenses, or exit.
    """
    print("Personal Expense Tracker (CLI)")

    while True:
        print("\nChoose an option:")
        print("1. Add Expense")
        print("2. View Expenses")
        print("3. Exit")
        choice = input("Enter choice (1/2/3): ").strip()

        if choice == '1':
            date = input("Enter date (YYYY-MM-DD) or leave blank for today: ").strip()
            if not date:
                date = datetime.today().strftime('%Y-%m-%d')

            category = input("Enter category: ").strip()
            while not category:
                category = input("Category cannot be empty. Enter category: ").strip()

            while True:
                amount_str = input("Enter amount: ").strip()
                try:
                    amount = float(amount_str)
```

```
                    break
                except ValueError:
                    print("Invalid amount. Please enter a number.")

            note = input("Enter note (optional): ").strip()
            save_expense(date, category, amount, note)

        elif choice == '2':
            filter_expenses()

        elif choice == '3':
            print("Goodbye!")
            break

        else:
            print("Invalid choice.")

if __name__ == "__main__":
    main()
```