Personal Expense Tracker: Course-End Project

1. Introduction and Problem Statement

In the modern, fast-paced world, effective personal finance management is crucial. This project addresses the need for a simple, yet robust, tool that allows individuals to monitor, categorize, and control their daily expenditures against a predefined monthly budget.

The objective was to design and implement a console-based **Personal Expense Tracker** application using Python. The tracker is built around core CRUD (Create, Read, Update, Delete) concepts and file handling, ensuring data persistence between sessions.

2. Project Objectives

The project successfully meets the following key objectives:

1. **Expense Management:** Enables users to log new daily expenses, capturing the date, category, amount, and a brief description.
2. **Categorization and Budgeting:** Allows users to set a specific monthly budget and tracks total spending against this limit, providing warnings or remaining balance updates.
3. **Data Persistence:** Implements robust file-handling functionality using the CSV format to save and load expense data, ensuring no data is lost upon program exit.
4. **User Experience:** Provides an intuitive, interactive, menu-driven command-line interface for ease of use.

3. Project Structure and Features

The application is structured into modular functions, each handling a specific core feature:

Function/Module	Description
`load_expenses()`	Reads existing expense data and the budget from a file (`expenses.csv`) upon application start.

`save_expenses()`	Writes all current expenses to the CSV file, overwriting the previous data.
`add_expense()`	Prompts the user for all necessary details (Date, Category, Amount, Description) and validates the inputs before storing the new expense.
`view_expenses()`	Displays all recorded expenses in a clear, formatted table structure for easy review.
`track_budget()`	Allows the user to set a monthly budget, calculates the total expenses, and reports the budget status (remaining balance or budget exceeded).
`main_menu()`	The core interactive loop that presents options to the user and calls the corresponding functions.

4. Execution and Output

A. Execution Steps

1. **Prerequisites:** Ensure Python is installed on your system.
2. **Save Code:** Save the provided Python code (`expense_tracker.py`) to a directory on your computer.

Run Application: Open your terminal or command prompt, navigate to the directory where you saved the file, and execute the command:
python expense_tracker.py

3.
4. **Interaction:** The main menu will appear. Enter the corresponding number (1-5) to perform an action. For example, enter `1` to add an expense, or `3` to set and track your budget.

B. Expected Workflow (Demonstration)

1. **Initial Load:** The program starts and attempts to load previous expenses from `expenses.csv`.
2. **Add Expense:** User selects `1`, enters details for a coffee purchase (e.g., `2024-12-05`, `Food`, `4.50`, `Morning coffee`). The expense is added to the list.
3. **Set Budget:** User selects `3`, sets the monthly budget to `1000.00`. The output confirms the remaining balance is `995.50`.
4. **Save & Exit:** User selects `5`. The application saves the current expenses to `expenses.csv` and terminates gracefully. The next time the program runs, it will load this saved data.