International Islamic University Chittagong

Department of CSE

Project Report

Course Title : Computer Algorithms Lab

Course code : CSE-2422

Personal Finance Budget Planner

Submitted to

Ms. Miskatul Jannat

Assistant Lecturer

Dept. of CSE,IIUC

Submitted by

Umme Samia Rahman Shose (C233454)

Tasnia Ayman Nuha (C233448)

Arifa Jahin (C233468)

Date of submission : 08/08/2025

Introduction

In today's fast-paced world, effective financial management is essential. The **"Personal Finance Budget Planner"** is a C++ based console application designed to help users manage and optimize their income, expenses, savings, and financial goals. It integrates well-known algorithmic strategies, such as greedy algorithms and dynamic programming, to provide smart suggestions for budgeting and achieving financial goals efficiently. This tool offers a user-friendly interface for inputting and categorizing expenses, enabling better visibility of spending patterns. By leveraging algorithmic techniques, it assists users in making data-driven decisions to maximize their financial benefits under given constraints.

Objective

- To develop a budget planning system that integrates algorithmic techniques for smarter financial decision-making.
- To enable users to efficiently track their income, expenses, and savings.
- To apply **Dynamic Programming** and **Greedy Algorithms** for optimized expense selection and goal scheduling.
- To assist users in planning financial goals effectively within a defined timeframe.
- To provide automated insights and categorized summaries for better financial clarity.

Background

Budgeting applications often rely on static data entry and simple arithmetic, lacking intelligent suggestions or planning features. This project addresses that gap by embedding classical algorithmic techniques into a real-world financial context. By applying the **0/1 Knapsack Algorithm**, the system can recommend the best combination of expenses under a given budget. Similarly, using a **Greedy approach to Job Scheduling**, it effectively prioritizes and schedules financial goals based on deadlines and benefits.

This blend of theory and practice not only strengthens algorithmic understanding but also demonstrates its application in solving everyday financial problems.

Hardware and Software Requirements

Hardware:

- **Processor:** Intel Core i3 or higher
- **RAM:** 4GB minimum
- **Storage:** 100MB free space

Software:

- **Operating System:** Compatible with Windows 10/11, Linux, or macOS.
- **IDE:** Code::Blocks , Dev C++ , VS Code
- **Language:** C++ with **compiler** GNU GCC
- **Text/File Handling:** Uses basic file I/O; no need for external libraries or databases.

System Design

The *Personal Finance Budget Planner* follows a modular, console-based architecture that integrates algorithmic decision-making with simple user interaction. Designed for clarity and efficiency, the system avoids external dependencies by using in-memory data structures and file-based persistence.

Key Components

1. **User Interface**
    - Console-based, menu-driven input for intuitive navigation and data entry.
2. **Data Management**
    - Uses structs and vectors to manage expenses, priorities, categories, and goals dynamically.
3. **Algorithm Engine**
    - **Knapsack (Dynamic Programming):** Selects optimal expense combinations within a set budget.
    - **Job Scheduling (Greedy Algorithm):** Prioritizes financial goals based on benefit and deadlines.
4. **Output Module**
    - Presents results in a structured console format and saves reports to .txt files for record-keeping.

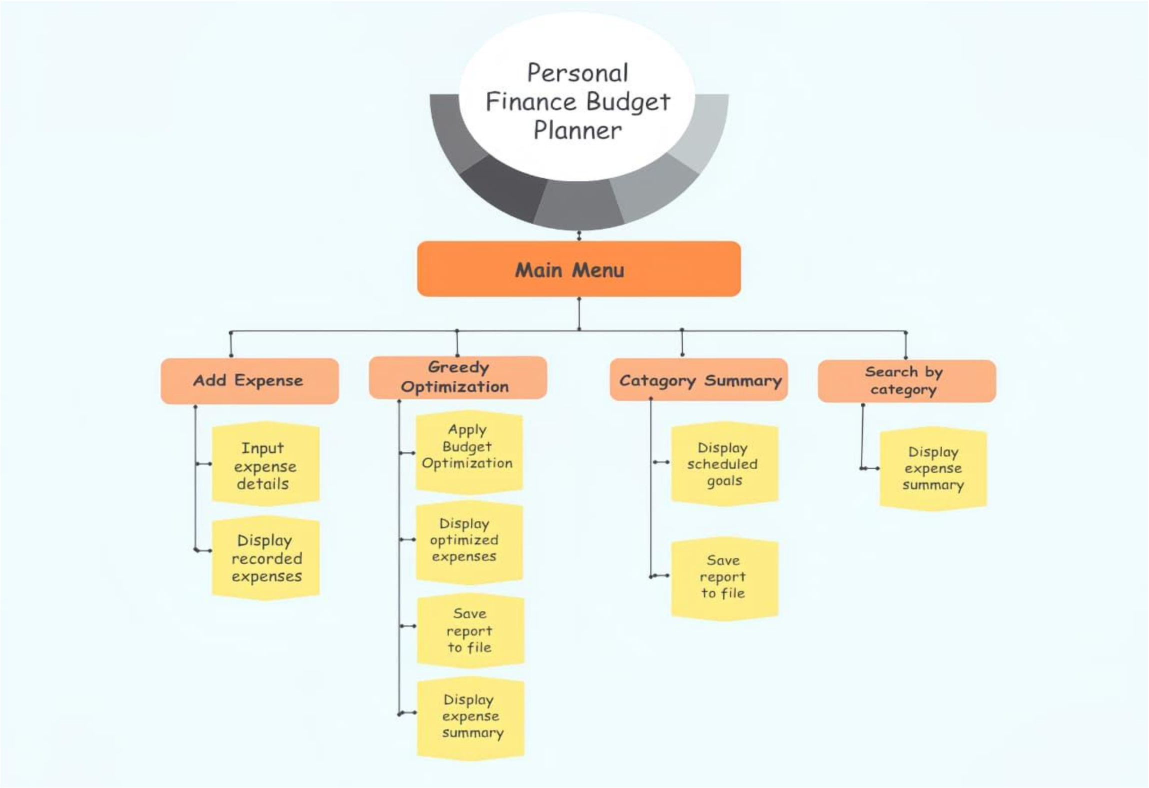
Design Highlights

- Lightweight and portable (runs on any OS with a C++ compiler).
- Modular structure allows for easy feature expansion.
- No external databases—relies on efficient file I/O for data storage.
- Built with clean code practices for maintainability.

Data Collection

Expense details, financial goals, and budget inputs are collected directly from users through a console-based interface. All data—such as cost, category, priority, and date—are stored in memory during runtime using structured data types, and persisted via text files for future reference.

Project Flow Diagram



Implementation Details

- **Language:** C++, compiled using GCC in Code::Blocks.
- **UI:** Console-based, menu-driven navigation.
- **Data Structures:** Used structs and vectors to store income, expenses, and financial goals.
- **Algorithms:**
  - ➢ **Knapsack (Dynamic Programming):** Selects the most valuable combination of expenses that fit within a given budget.
  - ➢ **Job Scheduling (Greedy):** Schedules financial goals by maximizing total benefit while respecting deadlines.
- **File Handling:** Saves reports in optimized_expenses.txt and scheduled_goals.txt.
- **Additional Features:**
  - ➢ Sorting and priority logic for goal selection.
  - ➢ Budget summary display with remaining balance.
  - ➢ Modular design for easy scalability.
- **Structs and Vectors**: Core C++ features used for data management.

Research Gap

Despite the growing need for personal budgeting tools, most existing solutions focus on GUI-based applications with static tracking features, lacking algorithmic intelligence. Very few console-based systems integrate **algorithmic optimization techniques** such as **Knapsack (Dynamic Programming)** or **Job Scheduling (Greedy Algorithm)** to support real-time decision-making. This gap limits their ability to provide smart financial recommendations based on user priorities and constraints.

This project addresses the gap by introducing an **algorithm-driven, console-based planner** that not only tracks expenses but also optimizes budget utilization and financial goal planning using classical algorithmic strategies.

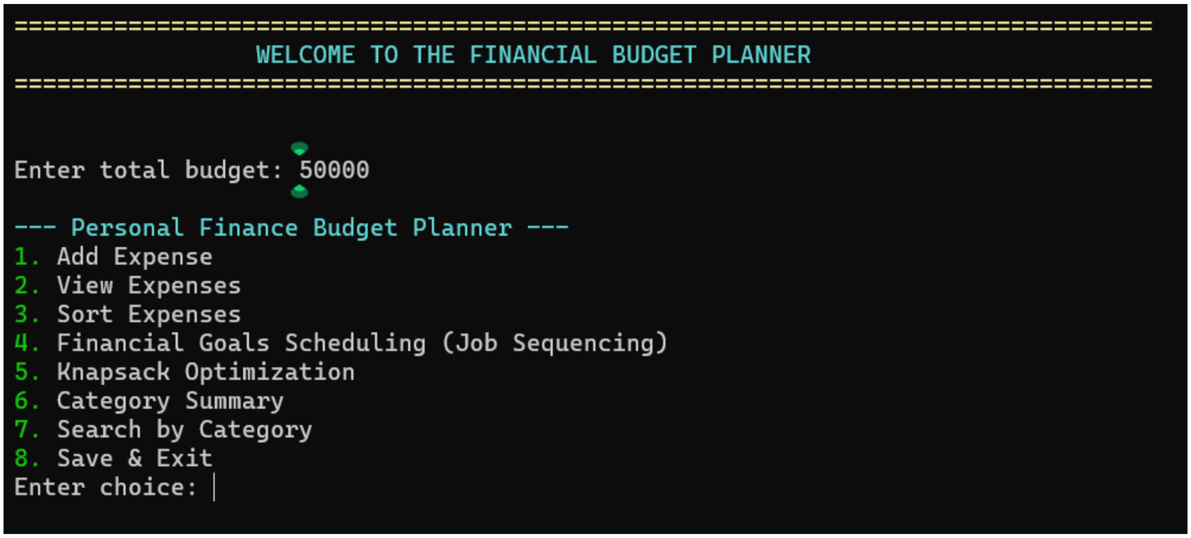
Relevance

This project highlights the practical value of combining **data structures** and **algorithm design** in developing efficient and intelligent budgeting systems.

Output Analysis

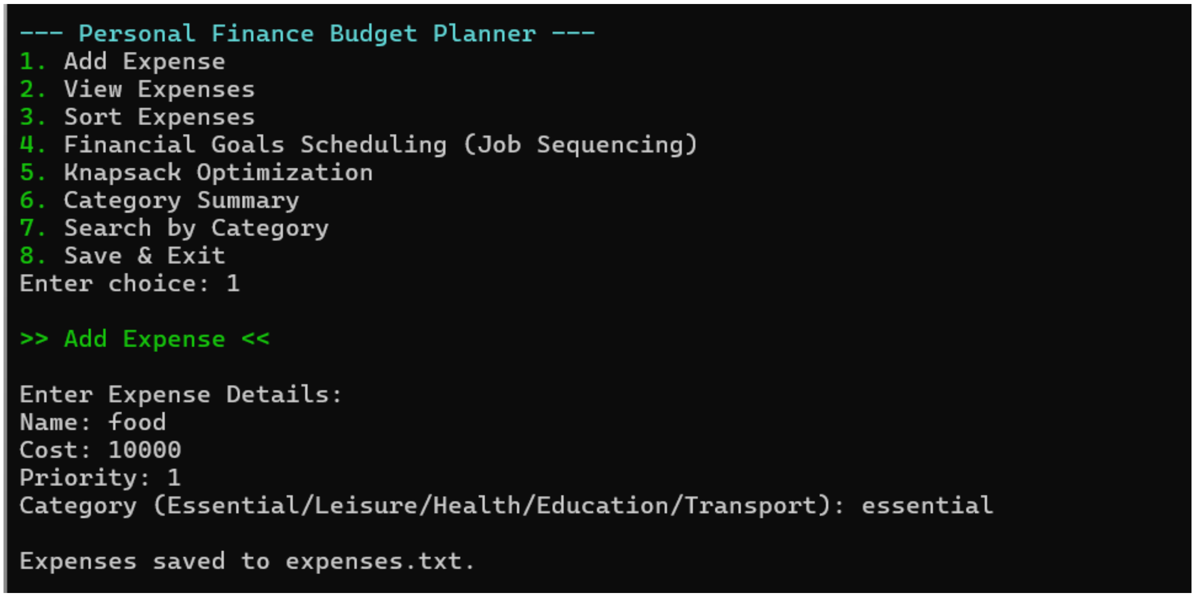
Console Output

- **User Main Menu**



```
================================================================================
                WELCOME TO THE FINANCIAL BUDGET PLANNER
================================================================================


Enter total budget: 50000

--- Personal Finance Budget Planner ---
1. Add Expense
2. View Expenses
3. Sort Expenses
4. Financial Goals Scheduling (Job Sequencing)
5. Knapsack Optimization
6. Category Summary
7. Search by Category
8. Save & Exit
Enter choice: 
```

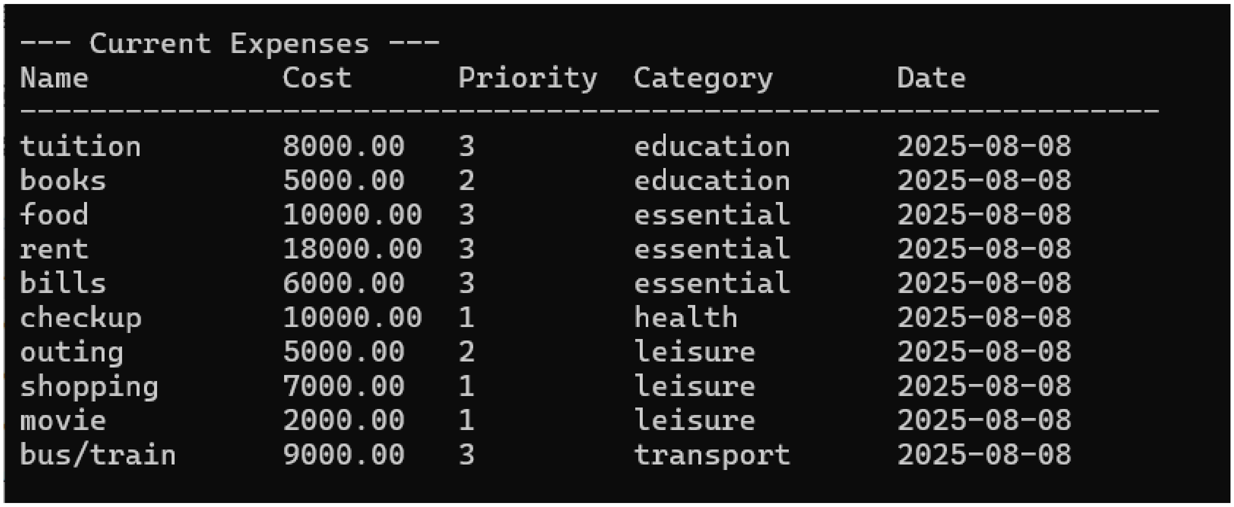
- **Adding expenses**



```
--- Personal Finance Budget Planner ---
1. Add Expense
2. View Expenses
3. Sort Expenses
4. Financial Goals Scheduling (Job Sequencing)
5. Knapsack Optimization
6. Category Summary
7. Search by Category
8. Save & Exit
Enter choice: 1

>> Add Expense <<

Enter Expense Details:
Name: food
Cost: 10000
Priority: 1
Category (Essential/Leisure/Health/Education/Transport): essential

Expenses saved to expenses.txt.
```

- User inputs detailed expense entries including name, cost,priority and category.

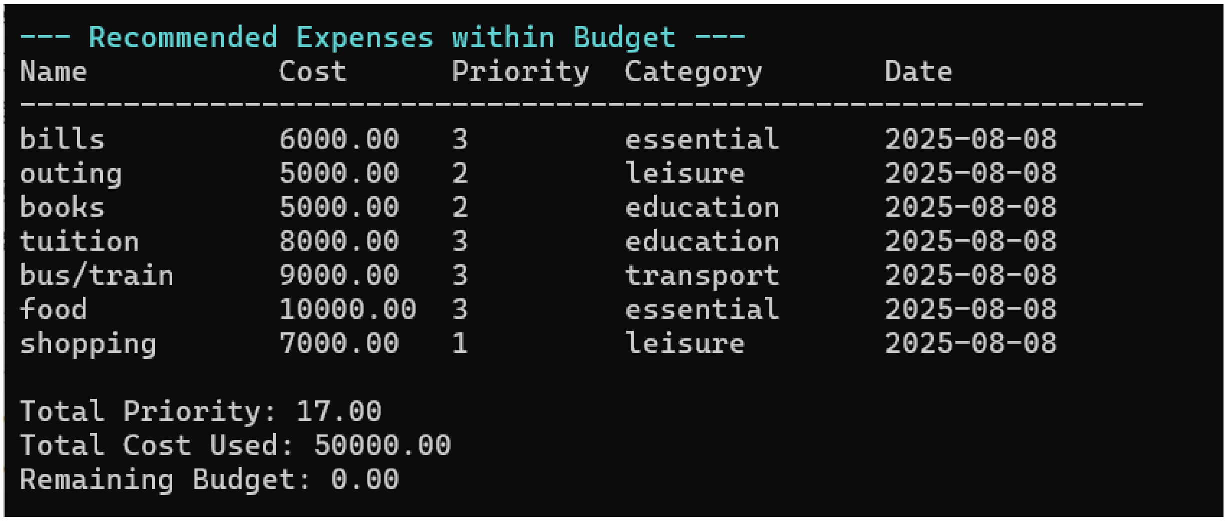
➢ Viewing and Sorting Expenses



```
--- Current Expenses ---
Name           Cost      Priority  Category       Date
-----------------------------------------------------------------
tuition        8000.00   3         education      2025-08-08
books          5000.00   2         education      2025-08-08
food           10000.00  3         essential      2025-08-08
rent           18000.00  3         essential      2025-08-08
bills          6000.00   3         essential      2025-08-08
checkup        10000.00  1         health         2025-08-08
outing         5000.00   2         leisure        2025-08-08
shopping       7000.00   1         leisure        2025-08-08
movie          2000.00   1         leisure        2025-08-08
bus/train      9000.00   3         transport      2025-08-08
```

- Displays all recorded expenses sorted by cost, priority, or category for better analysis.

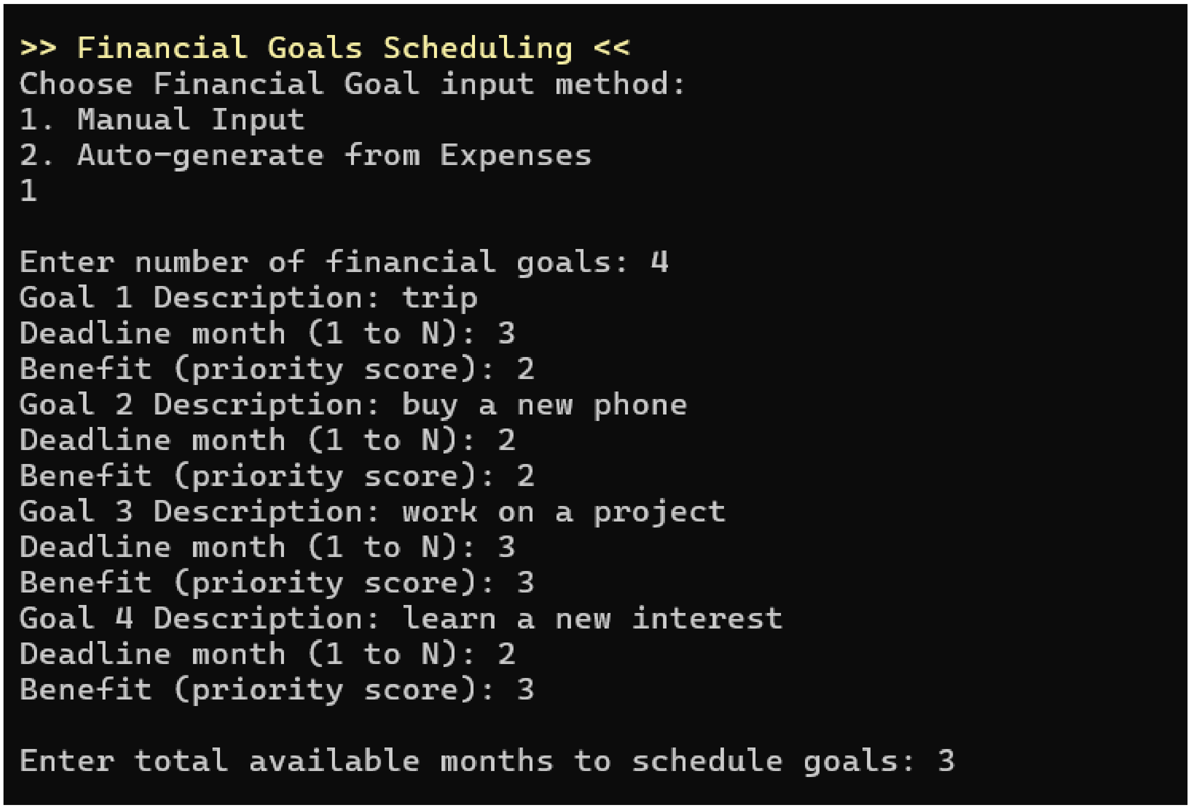
➢ Optimized Expense Selection using Knapsack Algorithm



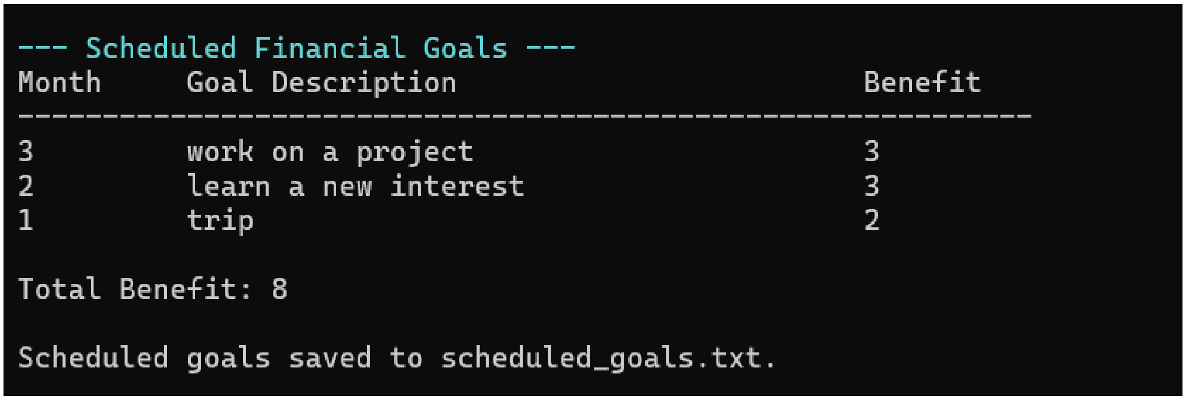
```
--- Recommended Expenses within Budget ---
Name           Cost      Priority  Category       Date
-----------------------------------------------------------------
bills          6000.00   3         essential      2025-08-08
outing         5000.00   2         leisure        2025-08-08
books          5000.00   2         education      2025-08-08
tuition        8000.00   3         education      2025-08-08
bus/train      9000.00   3         transport      2025-08-08
food           10000.00  3         essential      2025-08-08
shopping       7000.00   1         leisure        2025-08-08

Total Priority: 17.00
Total Cost Used: 50000.00
Remaining Budget: 0.00
```

- Selects the most valuable combination of expenses within a set budget using dynamic programming.

- **Financial Goal Scheduling using Greedy Algorithm**



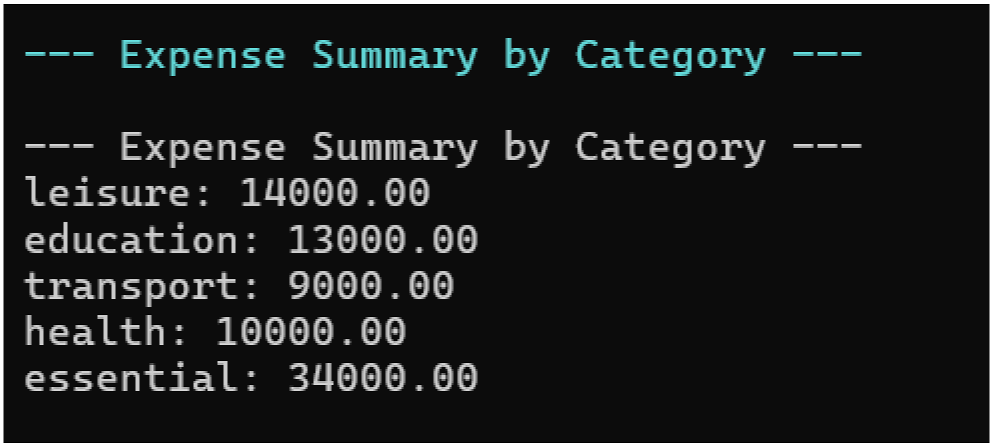
```
>> Financial Goals Scheduling <<
Choose Financial Goal input method:
1. Manual Input
2. Auto-generate from Expenses
1

Enter number of financial goals: 4
Goal 1 Description: trip
Deadline month (1 to N): 3
Benefit (priority score): 2
Goal 2 Description: buy a new phone
Deadline month (1 to N): 2
Benefit (priority score): 2
Goal 3 Description: work on a project
Deadline month (1 to N): 3
Benefit (priority score): 3
Goal 4 Description: learn a new interest
Deadline month (1 to N): 2
Benefit (priority score): 3

Enter total available months to schedule goals: 3
```



```
--- Scheduled Financial Goals ---
Month     Goal Description                        Benefit
------------------------------------------------------------
3         work on a project                       3
2         learn a new interest                    3
1         trip                                    2

Total Benefit: 8

Scheduled goals saved to scheduled_goals.txt.
```

- Schedules financial goals to maximize total benefit before their respective deadlines.

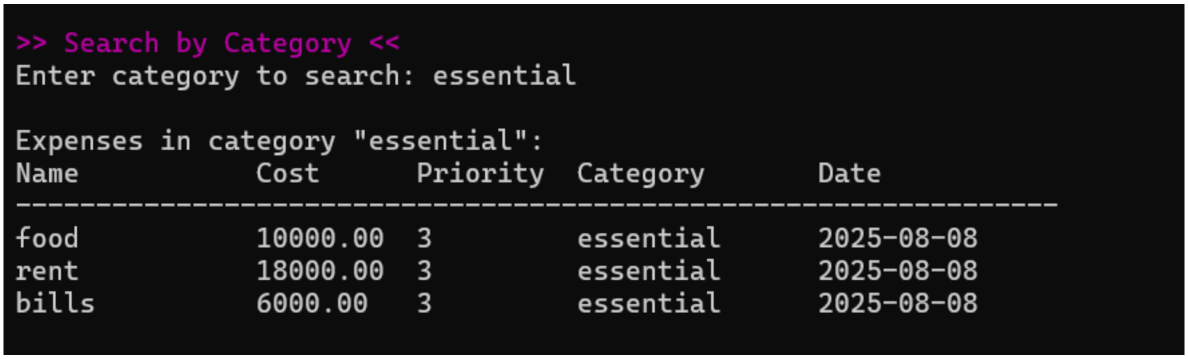
➢ Category-wise Expense Total



```
--- Expense Summary by Category ---

--- Expense Summary by Category ---
leisure: 14000.00
education: 13000.00
transport: 9000.00
health: 10000.00
essential: 34000.00
```

- Shows the total expense amount under a selected category for focused tracking.

➢ Search by Category

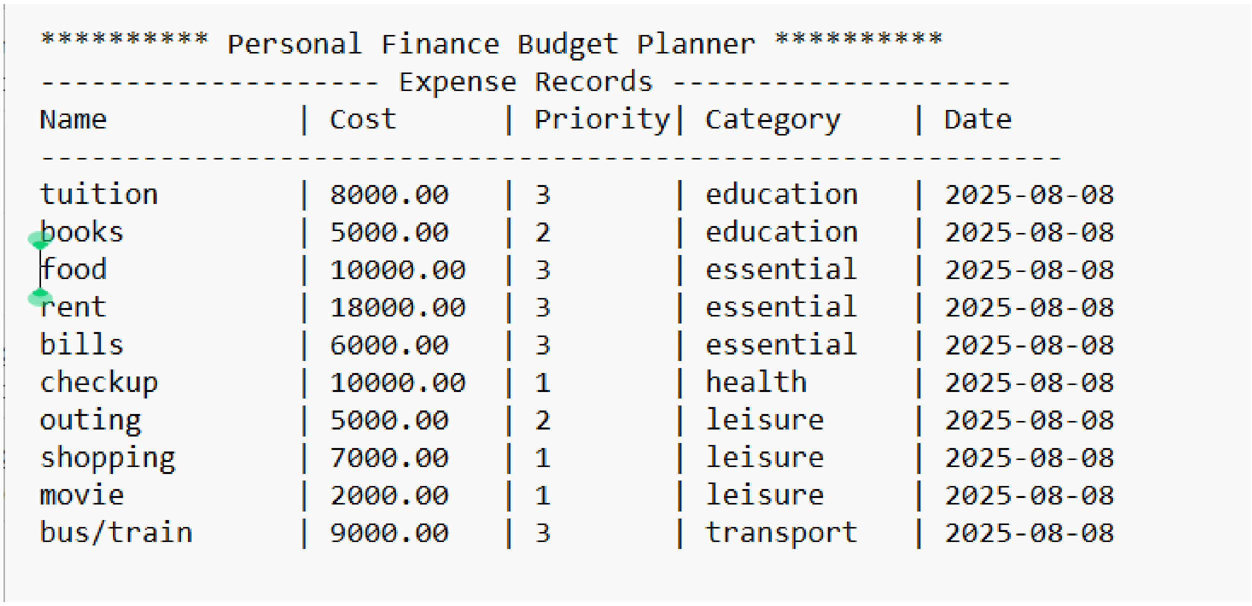


```
>> Search by Category <<
Enter category to search: essential

Expenses in category "essential":
Name           Cost      Priority  Category       Date
-----------------------------------------------------------------
food           10000.00  3         essential      2025-08-08
rent           18000.00  3         essential      2025-08-08
bills          6000.00   3         essential      2025-08-08
```

- Allows users to filter and view expenses belonging to a specific category.

Text Files

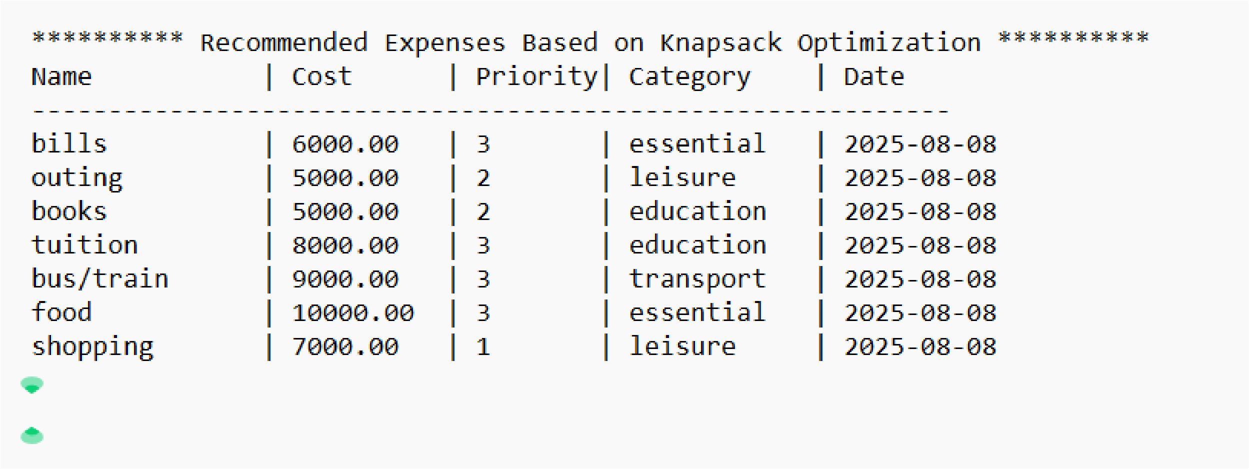
➢ All Entered Expenses (expenses.txt)



```
********** Personal Finance Budget Planner **********
-------------------- Expense Records --------------------
Name           | Cost      | Priority| Category    | Date
------------------------------------------------------------
tuition        | 8000.00   | 3       | education   | 2025-08-08
books          | 5000.00   | 2       | education   | 2025-08-08
food           | 10000.00  | 3       | essential   | 2025-08-08
rent           | 18000.00  | 3       | essential   | 2025-08-08
bills          | 6000.00   | 3       | essential   | 2025-08-08
checkup        | 10000.00  | 1       | health      | 2025-08-08
outing         | 5000.00   | 2       | leisure     | 2025-08-08
shopping       | 7000.00   | 1       | leisure     | 2025-08-08
movie          | 2000.00   | 1       | leisure     | 2025-08-08
bus/train      | 9000.00   | 3       | transport   | 2025-08-08
```

- Stores all expenses entered by the user with category, amount, and date for record keeping.

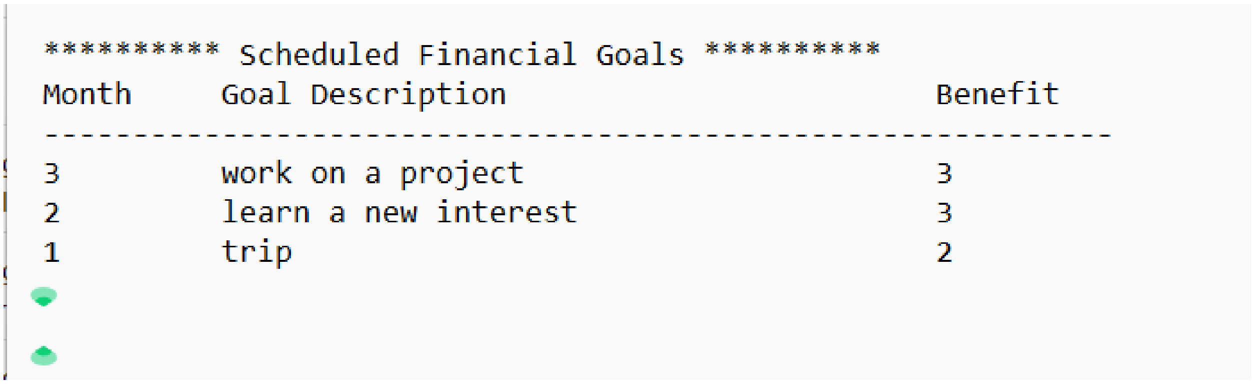
➢ Optimized Expenses Output (ecommended_expenses.txt)



```
********** Recommended Expenses Based on Knapsack Optimization **********
Name           | Cost      | Priority| Category    | Date
------------------------------------------------------------
bills          | 6000.00   | 3       | essential   | 2025-08-08
outing         | 5000.00   | 2       | leisure     | 2025-08-08
books          | 5000.00   | 2       | education   | 2025-08-08
tuition        | 8000.00   | 3       | education   | 2025-08-08
bus/train      | 9000.00   | 3       | transport   | 2025-08-08
food           | 10000.00  | 3       | essential   | 2025-08-08
shopping       | 7000.00   | 1       | leisure     | 2025-08-08
```

- Lists the selected set of expenses that maximize value under budget using the Knapsack algorithm.

- **Scheduled Financial Goals (scheduled_goals.txt)**



```
********** Scheduled Financial Goals **********
Month     Goal Description                        Benefit
------------------------------------------------------------
3         work on a project                       3
2         learn a new interest                    3
1         trip                                    2
```

- Contains a deadline-based priority list of financial goals selected using the Job Scheduling algorithm.

Challenges Faced and Future Scope

Challenges:

- **Limited Feature Set**: Focuses only on expense tracking and goal scheduling; lacks income tracking, authentication, and financial forecasting.
- **Scalability Issues**: Not optimized for handling large datasets or multi-user environments.
- **Console-Based Interface**: May not be user-friendly for non-technical users.
- **No Data Security**: Lacks encryption and user-level access control.
- **No Real-Time Integration**: Cannot sync with bank data or financial APIs.
- **Static Recommendations**: Suggestions are based on fixed algorithms, without learning or adapting to user behavior.

Future Scope:

- **Add GUI**: Develop a graphical interface to enhance user experience.
- **Implement User Accounts**: Introduce login, authentication, and multi-user support.
- **Integrate Cloud Storage**: Enable cloud syncing for cross-device access.
- **Incorporate Machine Learning**: Use ML to provide adaptive, personalized financial advice.
- **Develop Mobile App**: Create Android/iOS versions using Flutter or React Native.
- **Real-Time Expense Syncing**: Connect with financial APIs for live data integration.
- **Add Forecasting Tools**: Implement predictive models for income, savings, and spending.
- **Include Visual Analytics**: Add charts and graphs for better financial insights.
- **Enable Budget Alerts**: Notify users when spending approaches limits.
- **Expand Goal Types**: Allow recurring goals, custom time frames, and prioritization levels.

Conclusion

The project demonstrates how classic algorithmic strategies can be applied to real-world financial problems. With the inclusion of dynamic programming and greedy algorithms, the planner provides efficient and optimized suggestions. It’s an ideal base system for further development into a full-fledged financial planning tool.

References

- **GeeksforGeeks:** 0/1 Knapsack Problem
- **Programiz:** C++ File Handling
- **TutorialsPoint:** Greedy Algorithm for Job Scheduling
- **Stack Overflow and C++ documentation**

Appendix

CodeRepository:

Access the official GitHub repository here

Text Files Generated:

- Expenses.txt
- recommended_expenses.txt
- scheduled_goals.txt

Additional Diagrams: Available upon request or in project notebook.