Expense Tracker Application: Deliverable 2:

Core Functionality Implementation

Sri Sai Palamoor, Oishani Ganguly

Department of Computer and Information Science

University of the Cumberlands

Executive Master's in Computer Science

Professor Jay Thom

October 25, 2025

GitHub Repository: https://github.com/spalamoor39148/MSCS-632-M80-GroupProject

Overview

This is a cross-language CLI Expense Tracker implemented in both Python and C++. Both versions support adding, viewing, filtering, and deleting expenses; strict input validation (e.g., date format and positive amounts); summaries by category/total; and CSV/JSON import/export. The apps use persistent storage so changes survive restarts. It also includes automated tests in each language and clear run/test instructions. Directory layout separates `c++_impementation` and `python_implementation`, with a `deliverable_reports` folder for write-ups.

Python Implementation

The Python version lives under `python_implementation/expense_tracker` and exposes a menu-driven CLI. On startup it auto-loads from a persistent JSON file (`../expenses.json`) when present, so users can immediately continue where they left off. Core features include adding expenses with validated fields (date YYYY-MM-DD, positive numeric amount, valid category), listing and filtering by category or date range, showing a summary, and importing/exporting to CSV or JSON. Tests are `pytest`-based and cover both CLI flows and underlying logic (`tests/test_cli_features.py` and `tests/test_expense_manager.py`). This split encourages separation of concerns - business rules remain testable independently of the user interaction loop. From a usage

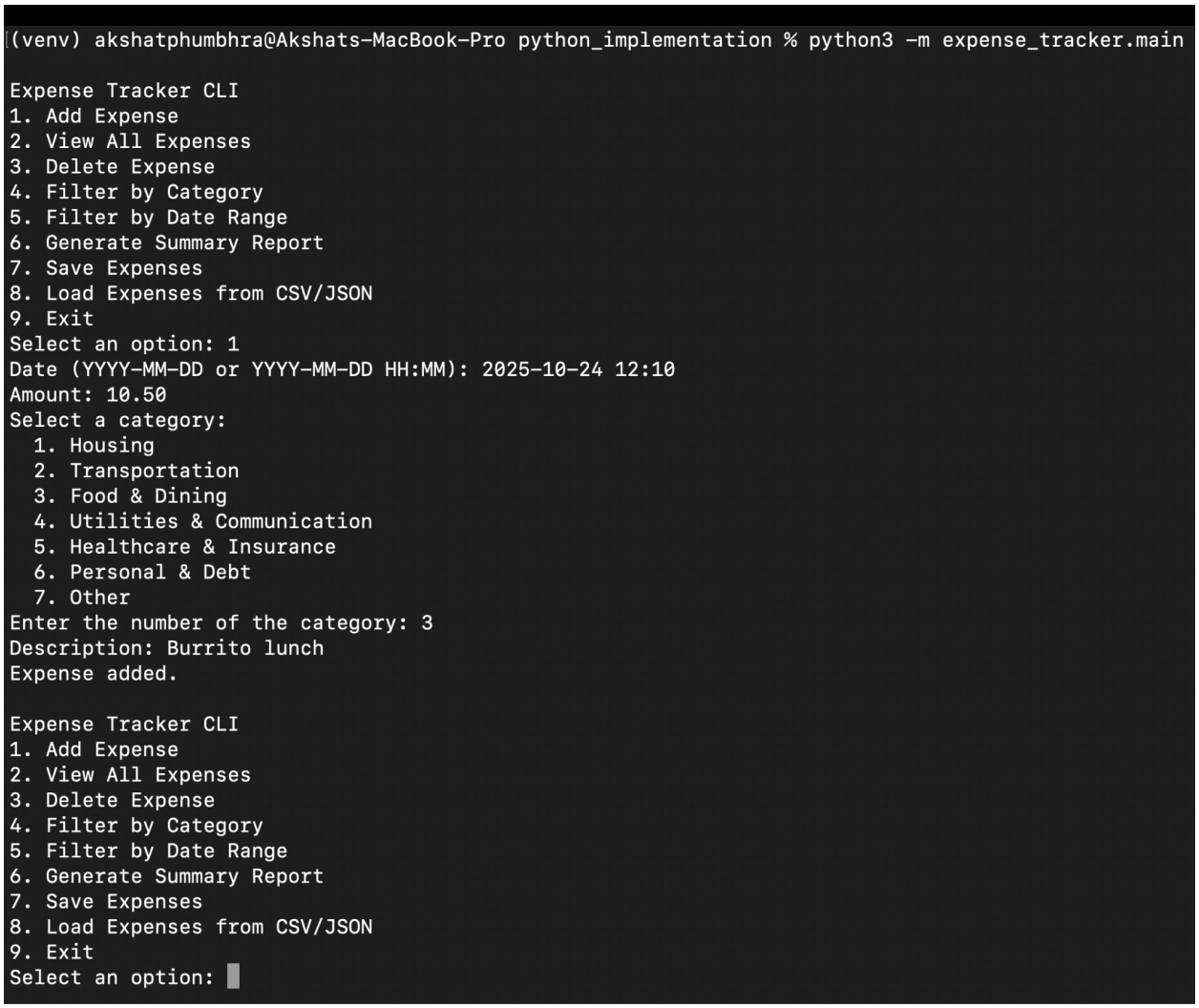
standpoint, running is simple: `cd python_implementation/expense_tracker && python3 main.py`, with tests invoked via `python3 -m pytest` from the `tests` directory.

C++ Implementation

The C++ version resides in `c++_impementation/expense_tracker_src` and is built with a provided `Makefile`. Like Python, it auto-loads a persistent CSV/JSON file (`expenses_persistent.csv/.json`) if available and keeps changes synced so sessions are stateful across runs. The user interacts with a textual menu to add, view, filter, summarize, delete, save, and load expenses, with strict input validation for dates and amounts. The project structure is modular: compilation/test commands reference `Expense.cpp`, `FileManager.cpp`, and `Utils.cpp`, reflecting a separation between the expense model, file I/O/persistence, and utility/validation helpers. A single, assert-based test binary exercises core features, edge cases, and persistence (built from `tests/test_expense_tracker.cpp` against the source files). This keeps the footprint minimal while ensuring the critical paths - CRUD, filters, summaries, and file round-trips - are covered. To run: `cd c++_impementation/expense_tracker_src && make && ./expense_tracker`; to test, a one-liner is provided that compiles the test harness and executes it.

Screenshots

Python Implementation



Add an expense manually

```
Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 1
Date (YYYY-MM-DD or YYYY-MM-DD HH:MM): 2025
Invalid date format. Please use YYYY-MM-DD or YYYY-MM-DD HH:MM (e.g., 2025-10-24 or 2025-10-24 14:30)
Date (YYYY-MM-DD or YYYY-MM-DD HH:MM): 2025-10-25
Amount: abc
Invalid amount. Please enter a positive number.
Amount: 8
Select a category:
  1. Housing
  2. Transportation
  3. Food & Dining
  4. Utilities & Communication
  5. Healthcare & Insurance
  6. Personal & Debt
  7. Other
Enter the number of the category: 2
Description:
This field cannot be empty.
Description: Tolls
Expense added.
```

Date, amount, and description input validation

```
Select a category:
  1. Housing
  2. Transportation
  3. Food & Dining
  4. Utilities & Communication
  5. Healthcare & Insurance
  6. Personal & Debt
  7. Other
Enter the number of the category: 9
Please enter a number between 1 and 7.
Enter the number of the category: 5
```

Category input validation

```
Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 2

ID | Date                | Amount   | Category               | Description
-------------------------------------------------------------------------------
1  | 2025-10-24 12:10    | $10.50   | Food & Dining          | Burrito lunch
2  | 2025-10-25 00:00    | $8.00    | Transportation         | Tolls
3  | 2025-10-24 14:30    | $3.00    | Food & Dining          | Hash browns at McD
4  | 2025-10-25 00:00    | $200.00  | Healthcare & Insurance | Medical consult
```

View all expenses

```
Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 4
Select a category:
  1. Housing
  2. Transportation
  3. Food & Dining
  4. Utilities & Communication
  5. Healthcare & Insurance
  6. Personal & Debt
  7. Other
Enter the number of the category: 3

ID | Date                | Amount   | Category               | Description
-------------------------------------------------------------------------------
1  | 2025-10-24 12:10    | $10.50   | Food & Dining          | Burrito lunch
2  | 2025-10-24 14:30    | $3.00    | Food & Dining          | Hash browns at McD
```

Filter by category

```
Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 5
Start date (YYYY-MM-DD or YYYY-MM-DD HH:MM): 2025-09-01
End date (YYYY-MM-DD or YYYY-MM-DD HH:MM): 2025-09-02
No expenses found.

Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 5
Start date (YYYY-MM-DD or YYYY-MM-DD HH:MM): 2025-10-23
End date (YYYY-MM-DD or YYYY-MM-DD HH:MM): 2025-10-24 13:30

ID | Date                | Amount   | Category               | Description
-------------------------------------------------------------------------------
1  | 2025-10-24 12:10     | $10.50    | Food & Dining          | Burrito lunch
```

Filter by date range: range with no expenses, range with expenses

```
Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 6

Total expenses: $221.50
By category:
  Housing: $0.00
  Transportation: $8.00
  Food & Dining: $13.50
  Utilities & Communication: $0.00
  Healthcare & Insurance: $200.00
  Personal & Debt: $0.00
  Other: $0.00
```

Summary report with total and per category format

```
Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 11
Invalid option. Try again.

Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 3

ID | Date                | Amount   | Category               | Description
-------------------------------------------------------------------------------
1  | 2025-10-24 12:10    | $10.50   | Food & Dining          | Burrito lunch
2  | 2025-10-25 00:00    | $8.00    | Transportation         | Tolls
3  | 2025-10-24 14:30    | $3.00    | Food & Dining          | Hash browns at McD
4  | 2025-10-25 00:00    | $200.00  | Healthcare & Insurance | Medical consult
Enter the ID of the expense to delete: 2
Deleted expense: Tolls on 2025-10-25 00:00.

Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 2

ID | Date                | Amount   | Category               | Description
-------------------------------------------------------------------------------
1  | 2025-10-24 12:10    | $10.50   | Food & Dining          | Burrito lunch
2  | 2025-10-24 14:30    | $3.00    | Food & Dining          | Hash browns at McD
3  | 2025-10-25 00:00    | $200.00  | Healthcare & Insurance | Medical consult
```

Invalid menu option validation, expense deletion by ID, view all expenses with deleted expense shown as removed

```
(venv) akshatphumbhra@Akshats-MacBook-Pro python_implementation % python3 -m expense_tracker.main

Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 7
Expenses saved.

Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 9
Goodbye!
(venv) akshatphumbhra@Akshats-MacBook-Pro python_implementation % python3 -m expense_tracker.main

Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 2

ID | Date                | Amount   | Category                | Description
-------------------------------------------------------------------------------
1  | 2025-10-24           | $10.50    | Food & Dining          | Burrito lunch
2  | 2025-10-24           | $3.00     | Food & Dining          | Hash browns at McD
3  | 2025-10-25           | $200.00   | Healthcare & Insurance | Medical consult
```

Save current expenses, exit, and then viewing all expenses again shows persistence of expenses

```
Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 8
Enter file path to load (CSV or JSON): ../input_data_samples/expenses_sample.csv
Loaded 8 valid expenses from ../input_data_samples/expenses_sample.csv.

Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 2

ID | Date                | Amount   | Category                | Description
-------------------------------------------------------------------------------
1  | 2025-10-24          | $10.50    | Food & Dining          | Burrito lunch
2  | 2025-10-24          | $3.00     | Food & Dining          | Hash browns at McD
3  | 2025-10-25          | $200.00   | Healthcare & Insurance | Medical consult
4  | 2025-10-24 08:15    | $12.50    | Food & Dining          | Lunch at cafe
5  | 2025-10-23 09:00    | $7.00     | Transportation         | Bus ticket
6  | 2025-10-22          | $20.00    | Utilities & Communication | Weekly shopping
7  | 2025-10-24 14:30    | $5.00     | Food & Dining          | Coffee
8  | 2025-10-21 19:00    | $1500.00  | Housing                | October rent
9  | 2025-10-20 10:00    | $200.00   | Healthcare & Insurance | Doctor visit
10 | 2025-10-19 16:00    | $100.00   | Personal & Debt        | Credit card payment
11 | 2025-10-18 12:00    | $50.00    | Other                  | Gift
```

```
Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 8
Enter file path to load (CSV or JSON): ../input_data_samples/expenses_sample.json
Loaded 8 valid expenses from ../input_data_samples/expenses_sample.json.

Expense Tracker CLI
1. Add Expense
2. View All Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Generate Summary Report
7. Save Expenses
8. Load Expenses from CSV/JSON
9. Exit
Select an option: 2

ID | Date                | Amount   | Category                | Description
-------------------------------------------------------------------------------
1  | 2025-10-24          | $10.50   | Food & Dining           | Burrito lunch
2  | 2025-10-24          | $3.00    | Food & Dining           | Hash browns at McD
3  | 2025-10-25          | $200.00  | Healthcare & Insurance  | Medical consult
4  | 2025-10-24 08:15    | $12.50   | Food & Dining           | Lunch at cafe
5  | 2025-10-23 09:00    | $7.00    | Transportation          | Bus ticket
6  | 2025-10-22          | $20.00   | Utilities & Communication | Weekly shopping
7  | 2025-10-24 14:30    | $5.00    | Food & Dining           | Coffee
8  | 2025-10-21 19:00    | $1500.00 | Housing                 | October rent
9  | 2025-10-20 10:00    | $200.00  | Healthcare & Insurance  | Doctor visit
10 | 2025-10-19 16:00    | $100.00  | Personal & Debt         | Credit card payment
11 | 2025-10-18 12:00    | $50.00   | Other                   | Gift
12 | 2025-10-24 08:15    | $12.50   | Food & Dining           | Lunch at cafe
13 | 2025-10-23 09:00    | $7.00    | Transportation          | Bus ticket
14 | 2025-10-22          | $20.00   | Utilities & Communication | Weekly shopping
15 | 2025-10-24 14:30    | $5.00    | Food & Dining           | Coffee
16 | 2025-10-21 19:00    | $1500.00 | Housing                 | October rent
17 | 2025-10-20 10:00    | $200.00  | Healthcare & Insurance  | Doctor visit
18 | 2025-10-19 16:00    | $100.00  | Personal & Debt         | Credit card payment
19 | 2025-10-18 12:00    | $50.00   | Other                   | Gift
```

Load expenses from an existing CSV or JSON file - combines the preloaded expenses with the manually entered expenses

C++ Implementation

```
✅ Expenses loaded from expenses_persistent.csv

====================================
         Expense Tracker CLI
====================================

========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 2

---------------- ALL EXPENSES ----------------
(Current data file: expenses_persistent.csv)
ID   Date        Amount      Category                Description
-------------------------------------------------------------
1    2025-10-24  12.00       Housing                 Gas bill
2    2025-10-24  12.00       Food & Dining           lunch
```

Preloading persistent expenses from preview sessions

```
========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 1
Enter date (YYYY-MM-DD): 2025-10-03
Enter amount: 12.60

Available Categories:
1. Housing
2. Transportation
3. Food & Dining
4. Utilities & Communication
5. Healthcare & Insurance
6. Personal & Debt
7. Other
Select category by number (1-7): 3
Enter description: Pizza lunch

✅ Expense added successfully! (ID: 3)
✅ Expenses saved to expenses_persistent.csv
```

Add an expense manually

```
========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 1
Enter date (YYYY-MM-DD): 24
❌ Invalid date format or value. Please enter a valid date in YYYY-MM-DD format.
Enter date (YYYY-MM-DD): 2025-10-24
Enter amount: abc
❌ Invalid amount. Please enter a positive number.
Enter amount: 12.a
❌ Invalid amount. Please enter a positive number.
Enter amount: 30.9

Available Categories:
1. Housing
2. Transportation
3. Food & Dining
4. Utilities & Communication
5. Healthcare & Insurance
6. Personal & Debt
7. Other
Select category by number (1-7): 9
Invalid number. Please select 1-7.
Select category by number (1-7): 6
Enter description: Shopping

✅ Expense added successfully! (ID: 4)
✅ Expenses saved to expenses_persistent.csv
```

Date, amount, and category input validation

```
========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 2

---------------- ALL EXPENSES ----------------
(Current data file: expenses_persistent.csv)
ID   Date        Amount      Category                  Description
-------------------------------------------------------------
1    2025-10-24  12.00       Housing                   Gas bill
2    2025-10-24  12.00       Food & Dining             lunch
3    2025-10-03  12.60       Food & Dining             Pizza lunch
4    2025-10-24  30.90       Personal & Debt           Shopping
```

View all expenses

```
========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 4

Available Categories:
1. Housing
2. Transportation
3. Food & Dining
4. Utilities & Communication
5. Healthcare & Insurance
6. Personal & Debt
7. Other
Select category number to filter: 3

Expenses in category: Food & Dining
2    2025-10-24  12.00       lunch
3    2025-10-03  12.60       Pizza lunch
```

Filter by category

```
========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 5
Enter start date (YYYY-MM-DD): 2026-10-09
Enter end date (YYYY-MM-DD): 2026-10-10

Expenses between 2026-10-09 and 2026-10-10:
No expenses found in this range.

========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 5
Enter start date (YYYY-MM-DD): 2025-10-02
Enter end date (YYYY-MM-DD): 2025-10-04

Expenses between 2025-10-02 and 2025-10-04:
3    2025-10-03  12.60       Food & Dining           Pizza lunch
```

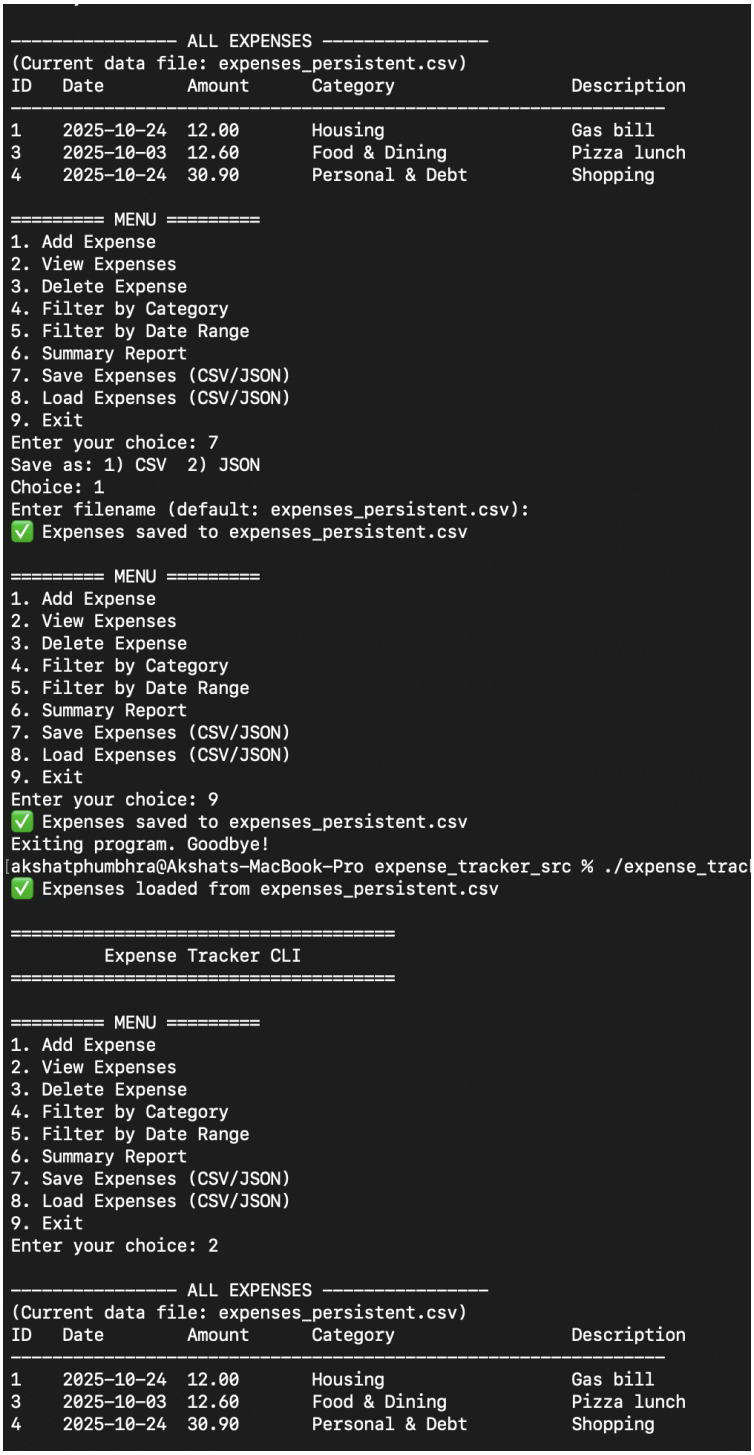
Filter by date range: range with no expenses, range with expenses

```
========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 6

------- Summary by Category -------
Personal & Debt          $30.90
Food & Dining            $24.60
Housing                  $12.00
-----------------------------------
Total                    $67.50
```

Summary report with total and per category format

```
========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 23
Invalid option. Please try again.

========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 3
Enter Expense ID to delete: 2
✅ Expense with ID 2 deleted successfully.
✅ Expenses saved to expenses_persistent.csv

========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 2

---------------- ALL EXPENSES ----------------
(Current data file: expenses_persistent.csv)
ID   Date        Amount      Category                Description
------------------------------------------------------------
1    2025-10-24  12.00       Housing                 Gas bill
3    2025-10-03  12.60       Food & Dining           Pizza lunch
4    2025-10-24  30.90       Personal & Debt         Shopping
```

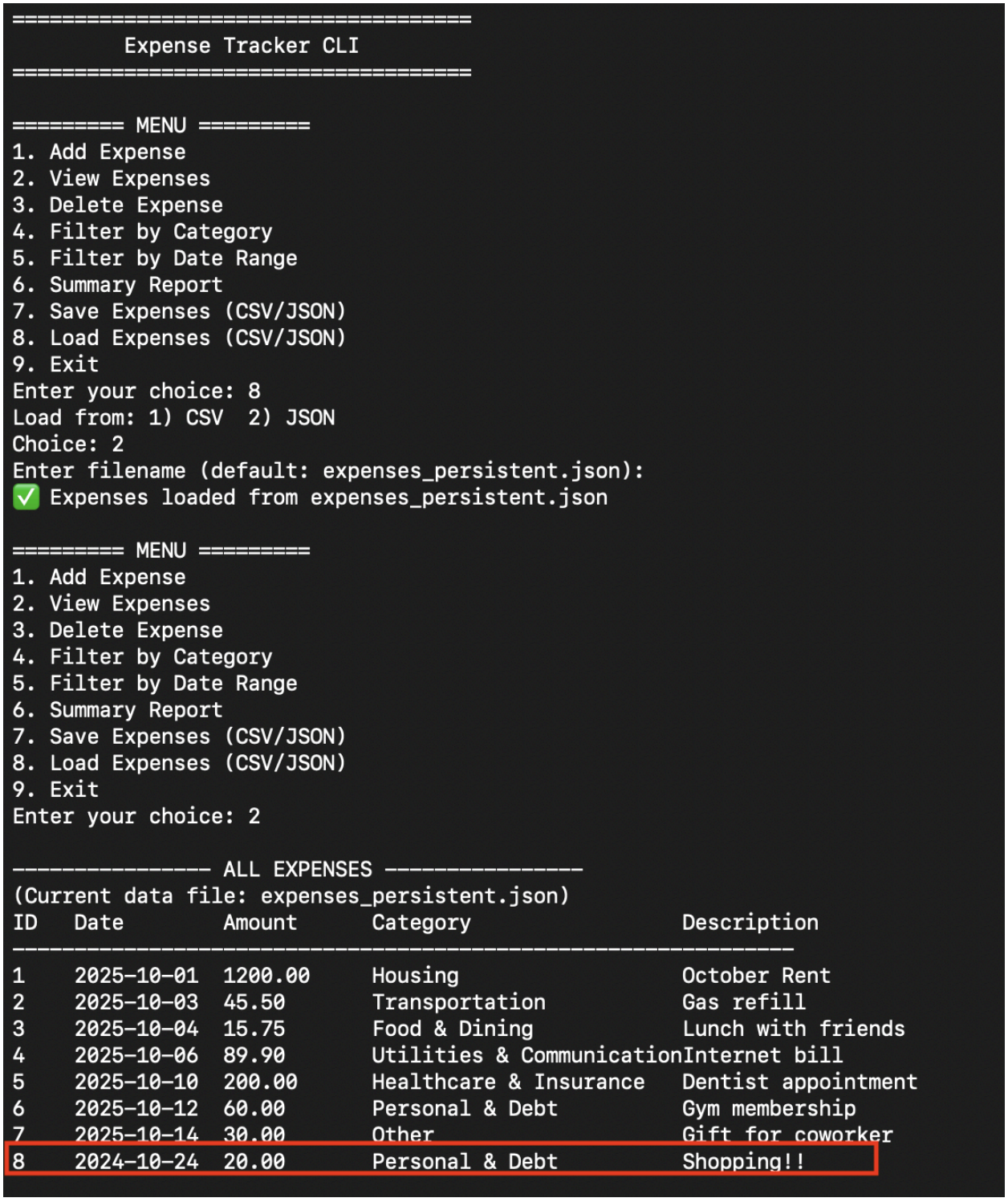
Invalid menu option validation, expense deletion by ID, view all expenses with deleted expense shown as removed



Save current expenses, exit, and then viewing all expenses again shows persistence of expenses

```
==================================
         Expense Tracker CLI
==================================

========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 8
Load from: 1) CSV  2) JSON
Choice: 1
Enter filename (default: expenses_persistent.csv): ../input_data_samples/expenses.csv
✅ Expenses loaded from ../input_data_samples/expenses.csv

========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 2

---------------- ALL EXPENSES ----------------
(Current data file: ../input_data_samples/expenses.csv)
ID   Date        Amount      Category                 Description
-----------------------------------------------------------------
1    2025-10-01  1200.00     Housing                  October Rent
2    2025-10-03  45.50       Transportation           Gas refill
3    2025-10-04  15.75       Food & Dining            Lunch with friends
4    2025-10-06  89.90       Utilities & CommunicationInternet bill
5    2025-10-10  200.00      Healthcare & Insurance   Dentist appointment
6    2025-10-12  60.00       Personal & Debt          Gym membership
7    2025-10-14  30.00       Other                    Gift for coworker
```

```
========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 8
Load from: 1) CSV  2) JSON
Choice: 2
Enter filename (default: expenses_persistent.json): ../input_data_samples/expenses.json
✅ Expenses loaded from ../input_data_samples/expenses.json

========= MENU =========
1. Add Expense
2. View Expenses
3. Delete Expense
4. Filter by Category
5. Filter by Date Range
6. Summary Report
7. Save Expenses (CSV/JSON)
8. Load Expenses (CSV/JSON)
9. Exit
Enter your choice: 2

---------------- ALL EXPENSES ----------------
(Current data file: ../input_data_samples/expenses.json)
ID   Date        Amount      Category                  Description
------------------------------------------------------------------
1    2025-10-01  1200.00     Housing                   October Rent
2    2025-10-03  45.50       Transportation            Gas refill
3    2025-10-04  15.75       Food & Dining             Lunch with friends
4    2025-10-06  89.90       Utilities & CommunicationInternet bill
5    2025-10-10  200.00      Healthcare & Insurance    Dentist appointment
6    2025-10-12  60.00       Personal & Debt           Gym membership
7    2025-10-14  30.00       Other                     Gift for coworker
```



Load expenses from an existing CSV or JSON file - combines the preloaded expenses with the manually entered expenses