Personal Expense Tracker

This document showcases a Python-based personal expense tracker. The program allows users to log expenses, categorize them, track spending against a monthly budget, and save data to a file for future use.

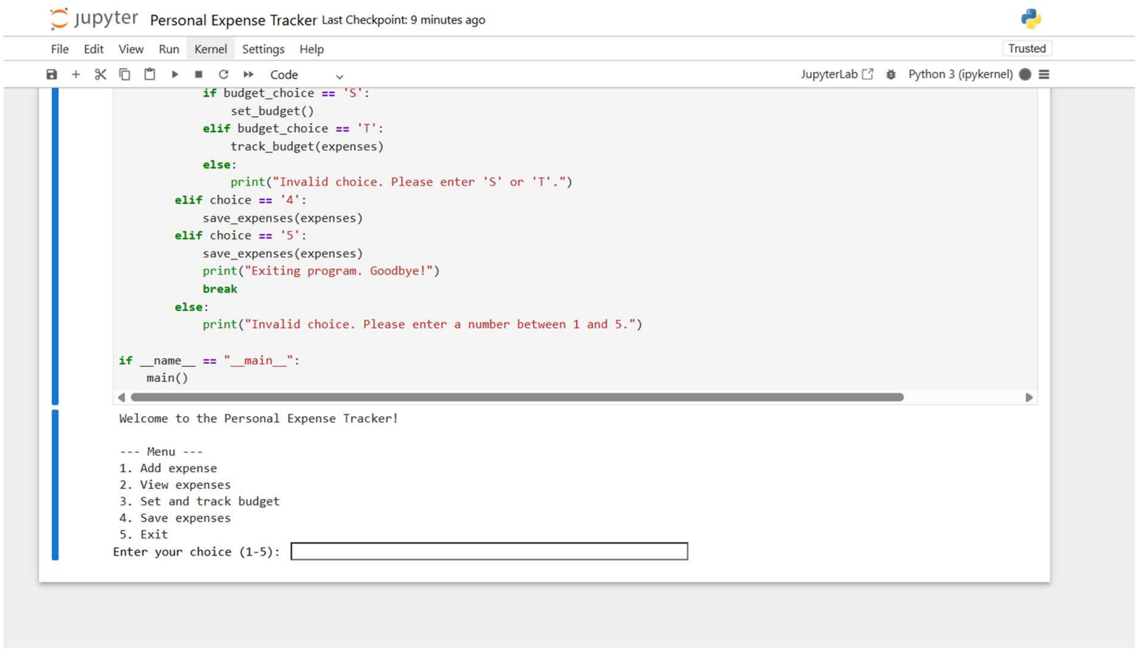
1. The Python Code

The following code implements all the core functionalities of the expense tracker, including adding, viewing, and saving expenses, as well as managing a monthly budget.

```
import csv

import datetime

import os


EXPENSES_FILE = 'expenses.csv'

BUDGET_FILE = 'budget.txt'


def add_expense(expenses):

    """Prompts the user for expense details and adds them to the expenses list."""

    date_str = input("Enter the date of the expense (YYYY-MM-DD): ")

    try:

        datetime.datetime.strptime(date_str, '%Y-%m-%d')

    except ValueError:

        print("Invalid date format. Please use YYYY-MM-DD.")

        return


    category = input("Enter the category of the expense (e.g., Food, Travel): ")

    try:

        amount = float(input("Enter the amount spent: "))

    except ValueError:

        print("Invalid amount. Please enter a number.")

        return
    description = input("Enter a brief description: ")

    new_expense = {
        'date': date_str,
        'category': category,
        'amount': amount,
        'description': description
    }
    expenses.append(new_expense)
    print("Expense added successfully!")

def view_expenses(expenses):
    """Displays all stored expenses."""
    if not expenses:
        print("No expenses to display.")
        return

    print("\n--- Your Expenses ---")
    for expense in expenses:
        if all(key in expense for key in ['date', 'category', 'amount', 'description']):
            print(f"Date: {expense['date']}, Category: {expense['category']}, Amount:
${expense['amount']:.2f}, Description: {expense['description']}")
        else:
            print("Incomplete expense record found. Skipping.")
    print("---------------------\n")

def set_budget():
    """Allows the user to set a monthly budget."""
    try:
        budget = float(input("Enter your total monthly budget: "))
        with open(BUDGET_FILE, 'w') as f:
            f.write(str(budget))
        print("Monthly budget set successfully!")
    except ValueError:
        print("Invalid budget amount. Please enter a number.")

def track_budget(expenses):
    """Calculates total expenses and compares them to the budget."""
    total_expenses = sum(expense['amount'] for expense in expenses)

    budget = 0.0
    if os.path.exists(BUDGET_FILE):
        with open(BUDGET_FILE, 'r') as f:
            try:
                budget = float(f.read())
            except ValueError:
                print("Invalid budget data in file. Please set a new budget.")
                return
    else:
        print("No budget set. Please set a budget first.")
        return

    print(f"\n--- Budget Tracker ---")
    print(f"Total expenses so far: ${total_expenses:.2f}")
    print(f"Monthly budget: ${budget:.2f}")

    remaining_balance = budget - total_expenses
    if remaining_balance < 0:
        print(f"You have exceeded your budget by ${-remaining_balance:.2f}!")
    else:
        print(f"You have ${remaining_balance:.2f} left for the month.")
    print("--------------------\n")


def save_expenses(expenses):
    """Saves all expenses to a CSV file."""
    with open(EXPENSES_FILE, 'w', newline='') as csvfile:
        fieldnames = ['date', 'category', 'amount', 'description']
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        writer.writeheader()
        writer.writerows(expenses)
    print("Expenses saved to expenses.csv.")


def load_expenses():
    """Loads expenses from a CSV file."""
    expenses = []
    if os.path.exists(EXPENSES_FILE):
        with open(EXPENSES_FILE, 'r') as csvfile:
            reader = csv.DictReader(csvfile)
            for row in reader:
                try:
                    row['amount'] = float(row['amount'])
                    expenses.append(row)
                except (ValueError, KeyError):
                    print(f"Skipping invalid row in CSV: {row}")
    return expenses


def main():
    """Main function to run the interactive menu."""
    expenses = load_expenses()
    print("Welcome to the Personal Expense Tracker!")

    while True:
        print("\n--- Menu ---")
        print("1. Add expense")
        print("2. View expenses")
        print("3. Set and track budget")
        print("4. Save expenses")
        print("5. Exit")
        choice = input("Enter your choice (1-5): ")

        if choice == '1':
            add_expense(expenses)
        elif choice == '2':
            view_expenses(expenses)
        elif choice == '3':
            budget_choice = input("Enter 'S' to set a budget or 'T' to track your budget: ").upper()
            if budget_choice == 'S':
                set_budget()
            elif budget_choice == 'T':
                track_budget(expenses)
            else:
                print("Invalid choice. Please enter 'S' or 'T'.")
        elif choice == '4':
            save_expenses(expenses)
        elif choice == '5':
            save_expenses(expenses)
            print("Exiting program. Goodbye!")
            break
        else:
            print("Invalid choice. Please enter a number between 1 and 5.")

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

2. Interactive Menu and User Interface

The program features a simple, menu-driven interface. When the user runs the script, they are presented with a list of options.

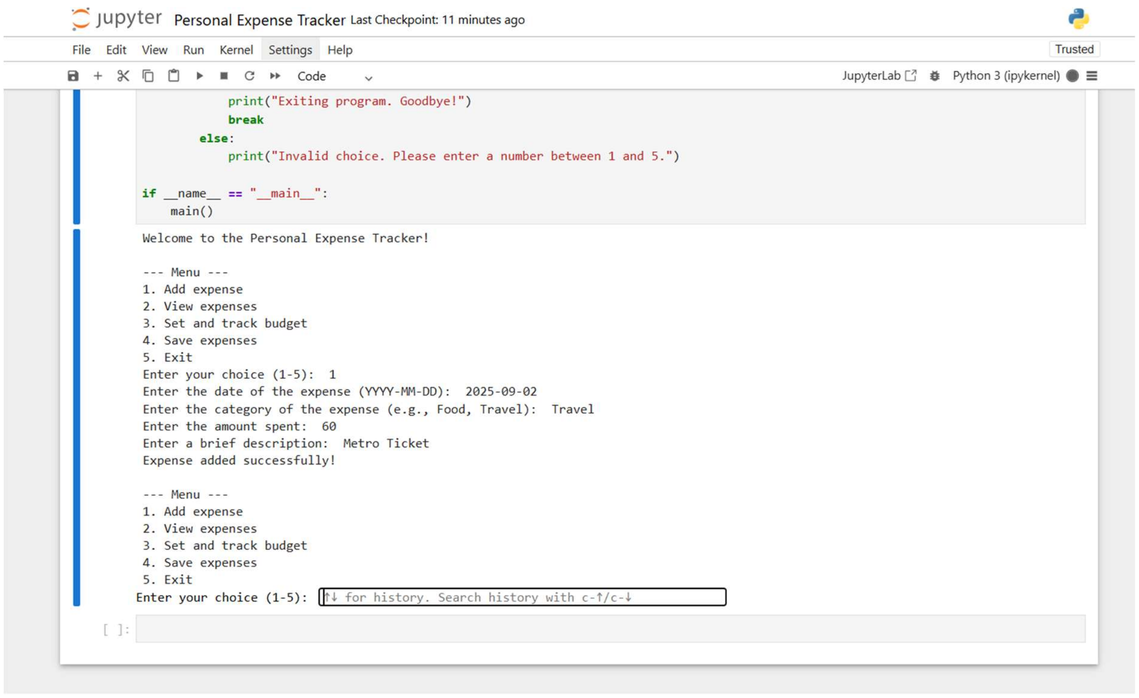

Jupyter Personal Expense Tracker Last Checkpoint: 9 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

```
            if budget_choice == 'S':
                set_budget()
            elif budget_choice == 'T':
                track_budget(expenses)
            else:
                print("Invalid choice. Please enter 'S' or 'T'.")
        elif choice == '4':
            save_expenses(expenses)
        elif choice == '5':
            save_expenses(expenses)
            print("Exiting program. Goodbye!")
            break
        else:
            print("Invalid choice. Please enter a number between 1 and 5.")

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

```
Welcome to the Personal Expense Tracker!

--- Menu ---
1. Add expense
2. View expenses
3. Set and track budget
4. Save expenses
5. Exit
Enter your choice (1-5):
```

Adding an Expense

By selecting option 1, the user can input the details for a new expense. The program prompts for the date, category, amount, and a description.

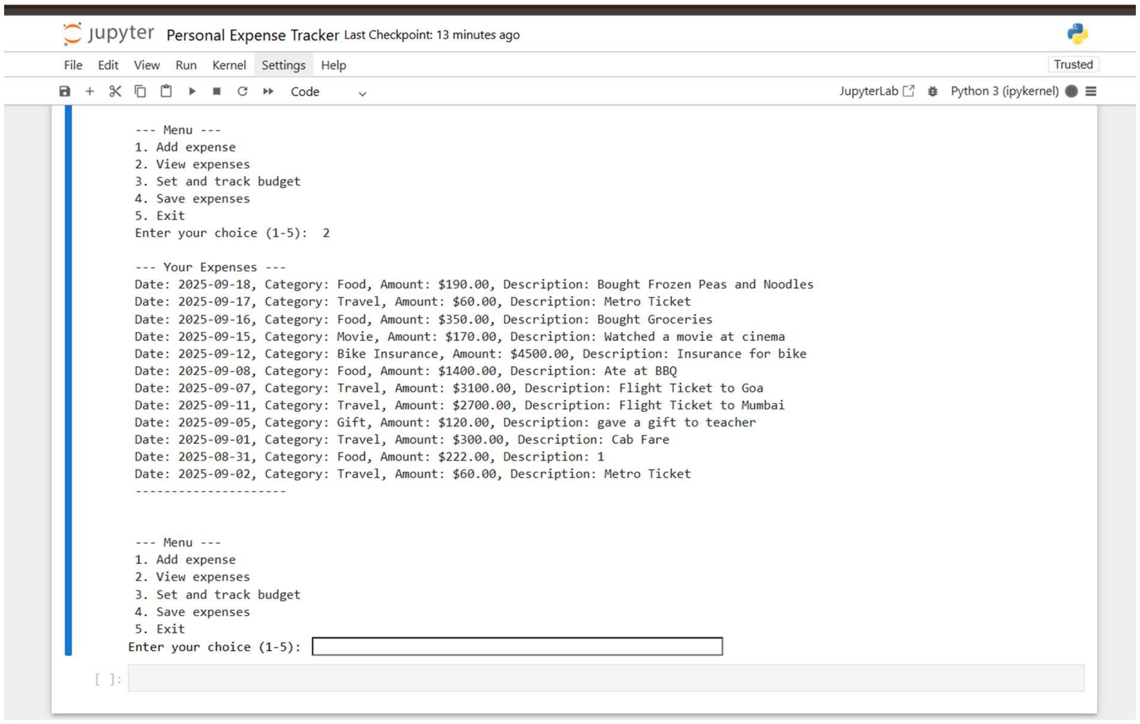

jupyter Personal Expense Tracker Last Checkpoint: 11 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

```
            print("Exiting program. Goodbye!")
            break
        else:
            print("Invalid choice. Please enter a number between 1 and 5.")

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

```
Welcome to the Personal Expense Tracker!

--- Menu ---
1. Add expense
2. View expenses
3. Set and track budget
4. Save expenses
5. Exit
Enter your choice (1-5):  1
Enter the date of the expense (YYYY-MM-DD):  2025-09-02
Enter the category of the expense (e.g., Food, Travel):  Travel
Enter the amount spent:  60
Enter a brief description:  Metro Ticket
Expense added successfully!

--- Menu ---
1. Add expense
2. View expenses
3. Set and track budget
4. Save expenses
5. Exit
Enter your choice (1-5):  ↑↓ for history. Search history with c-↑/c-↓
```

Viewing Expenses

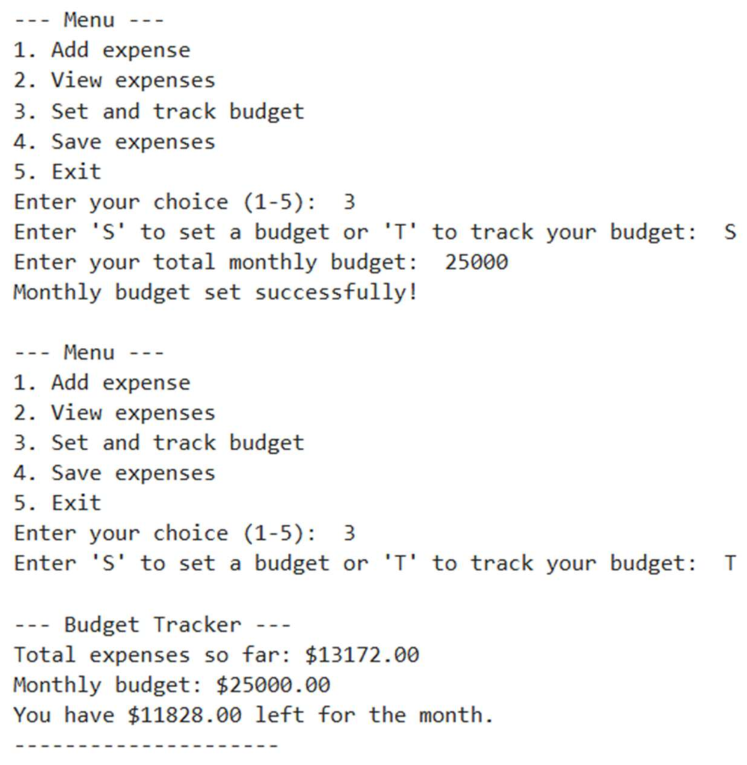
Option 2 displays all the expenses that have been recorded so far. It presents the data in a clear, easy-to-read format.


jupyter Personal Expense Tracker Last Checkpoint: 13 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

```
--- Menu ---
1. Add expense
2. View expenses
3. Set and track budget
4. Save expenses
5. Exit
Enter your choice (1-5):  2

--- Your Expenses ---
Date: 2025-09-18, Category: Food, Amount: $190.00, Description: Bought Frozen Peas and Noodles
Date: 2025-09-17, Category: Travel, Amount: $60.00, Description: Metro Ticket
Date: 2025-09-16, Category: Food, Amount: $350.00, Description: Bought Groceries
Date: 2025-09-15, Category: Movie, Amount: $170.00, Description: Watched a movie at cinema
Date: 2025-09-12, Category: Bike Insurance, Amount: $4500.00, Description: Insurance for bike
Date: 2025-09-08, Category: Food, Amount: $1400.00, Description: Ate at BBQ
Date: 2025-09-07, Category: Travel, Amount: $3100.00, Description: Flight Ticket to Goa
Date: 2025-09-11, Category: Travel, Amount: $2700.00, Description: Flight Ticket to Mumbai
Date: 2025-09-05, Category: Gift, Amount: $120.00, Description: gave a gift to teacher
Date: 2025-09-01, Category: Travel, Amount: $300.00, Description: Cab Fare
Date: 2025-08-31, Category: Food, Amount: $222.00, Description: 1
Date: 2025-09-02, Category: Travel, Amount: $60.00, Description: Metro Ticket
--------------------


--- Menu ---
1. Add expense
2. View expenses
3. Set and track budget
4. Save expenses
5. Exit
Enter your choice (1-5):
```

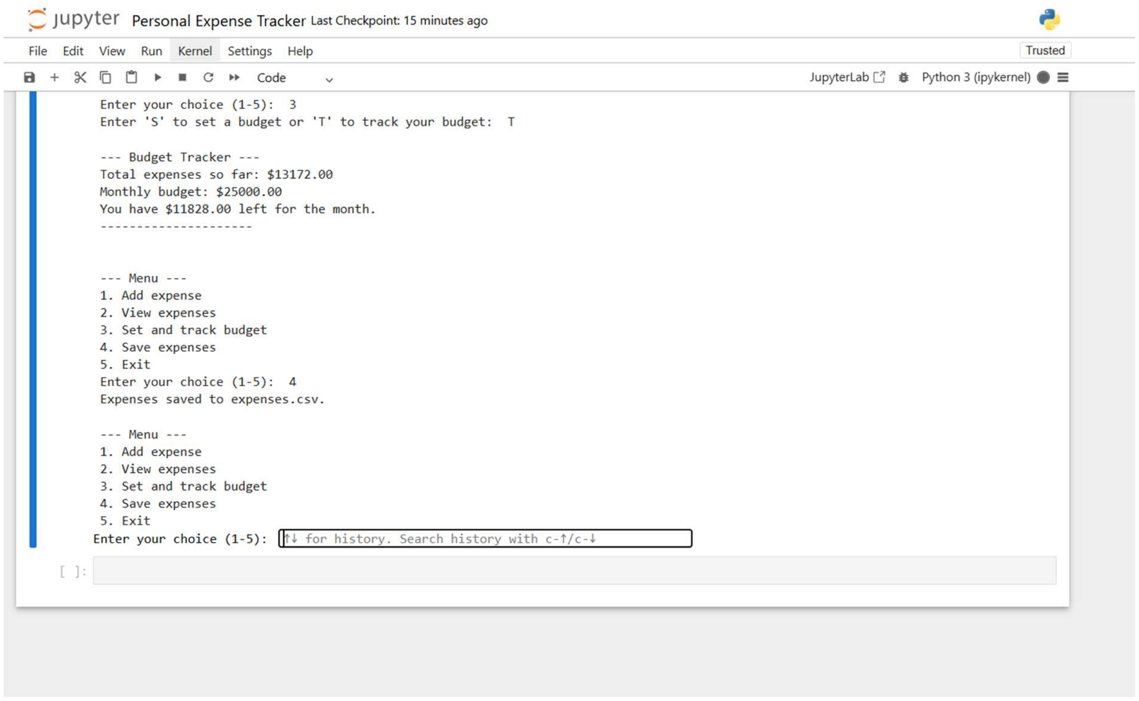
Setting and Tracking the Budget

When the user selects option 3, they are given the choice to either set a new budget or track their spending against the current budget.



```
--- Menu ---
1. Add expense
2. View expenses
3. Set and track budget
4. Save expenses
5. Exit
Enter your choice (1-5):  3
Enter 'S' to set a budget or 'T' to track your budget:  S
Enter your total monthly budget:  25000
Monthly budget set successfully!

--- Menu ---
1. Add expense
2. View expenses
3. Set and track budget
4. Save expenses
5. Exit
Enter your choice (1-5):  3
Enter 'S' to set a budget or 'T' to track your budget:  T

--- Budget Tracker ---
Total expenses so far: $13172.00
Monthly budget: $25000.00
You have $11828.00 left for the month.
---------------------
```

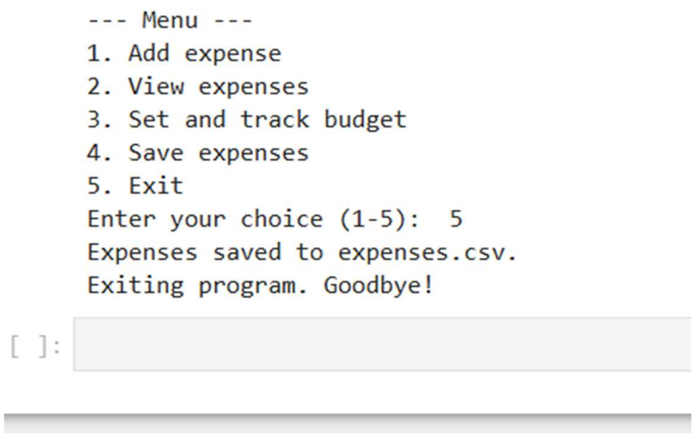
Saving Expenses



jupyter Personal Expense Tracker Last Checkpoint: 15 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

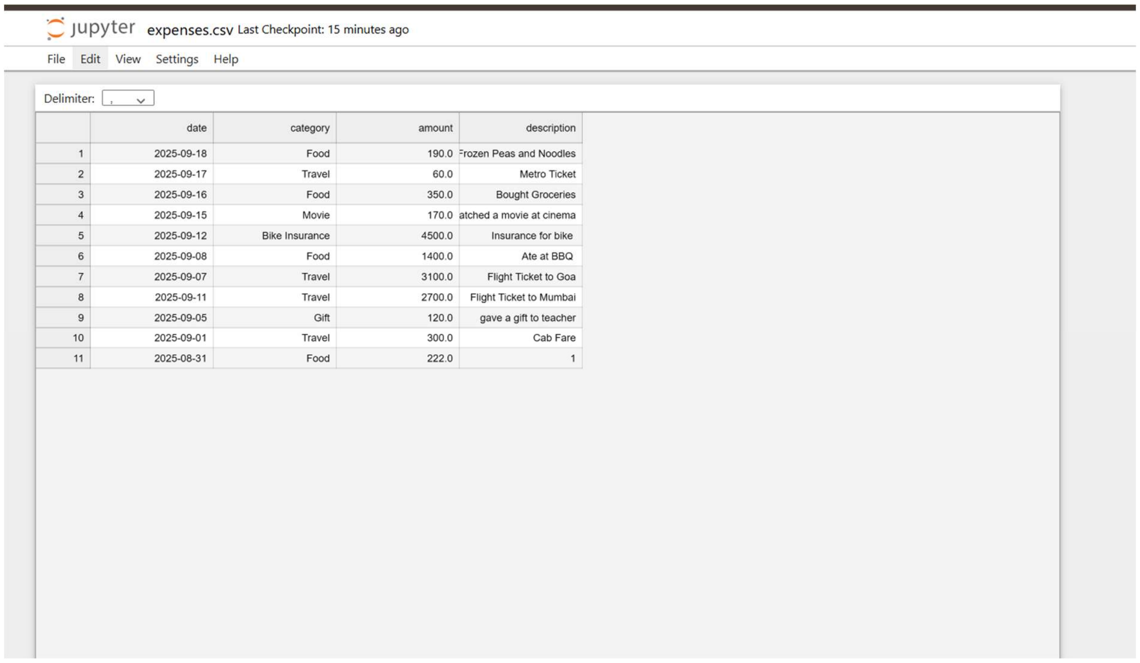
```
Enter your choice (1-5):  3
Enter 'S' to set a budget or 'T' to track your budget:  T

--- Budget Tracker ---
Total expenses so far: $13172.00
Monthly budget: $25000.00
You have $11828.00 left for the month.
--------------------


--- Menu ---
1. Add expense
2. View expenses
3. Set and track budget
4. Save expenses
5. Exit
Enter your choice (1-5):  4
Expenses saved to expenses.csv.

--- Menu ---
1. Add expense
2. View expenses
3. Set and track budget
4. Save expenses
5. Exit
Enter your choice (1-5):  ↑↓ for history. Search history with c-↑/c-↓

[ ]:
```

Exit Program



```
--- Menu ---
1. Add expense
2. View expenses
3. Set and track budget
4. Save expenses
5. Exit
Enter your choice (1-5):  5
Expenses saved to expenses.csv.
Exiting program. Goodbye!

[ ]:
```

Express.csv data



expenses.csv Last Checkpoint: 15 minutes ago

Delimiter: ,

	date	category	amount	description
1	2025-09-18	Food	190.0	rozen Peas and Noodles
2	2025-09-17	Travel	60.0	Metro Ticket
3	2025-09-16	Food	350.0	Bought Groceries
4	2025-09-15	Movie	170.0	atched a movie at cinema
5	2025-09-12	Bike Insurance	4500.0	Insurance for bike
6	2025-09-08	Food	1400.0	Ate at BBQ
7	2025-09-07	Travel	3100.0	Flight Ticket to Goa
8	2025-09-11	Travel	2700.0	Flight Ticket to Mumbai
9	2025-09-05	Gift	120.0	gave a gift to teacher
10	2025-09-01	Travel	300.0	Cab Fare
11	2025-08-31	Food	222.0	1