JULY 23, 2024

PYTHON

EXPENSE TRACKER

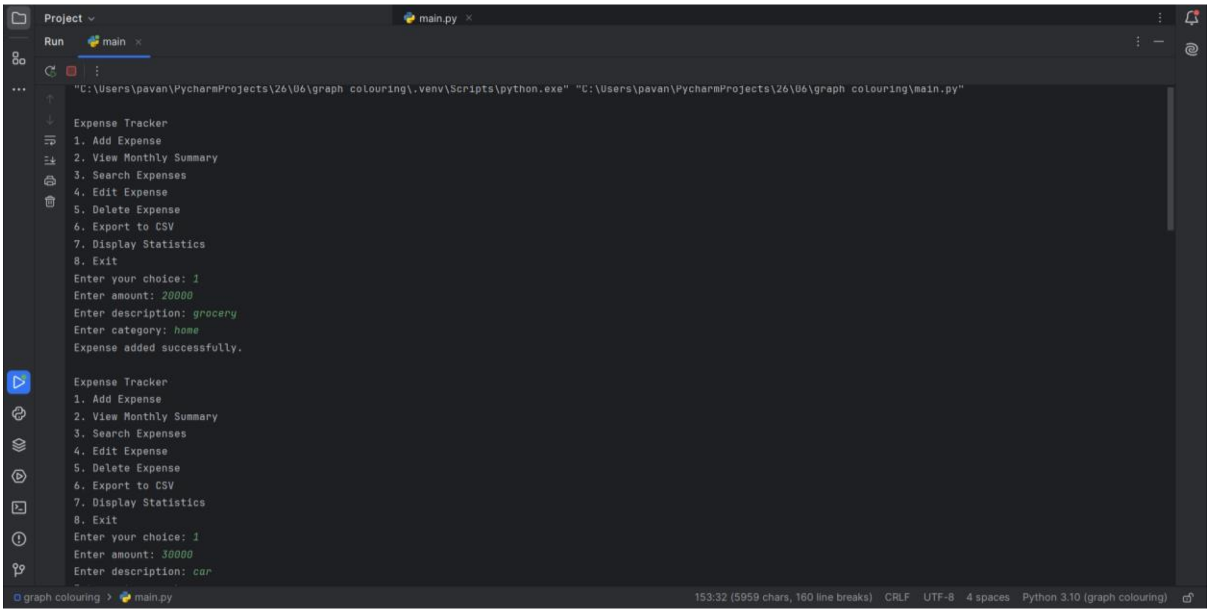
PRISHA D

CODE:

```
import json
from datetime import datetime
import os
import csv

# Define file to store expenses
FILE_PATH = 'expenses.json'

# Load existing data or create new
def load_data():
    """Load existing expense data from a JSON file or create an empty list if the file doesn't exist."""
    if os.path.exists(FILE_PATH):
        with open(FILE_PATH, 'r') as file:
            return json.load(file)
    return []

# Save data to file
def save_data(data):
    """Save the current expense data to a JSON file."""
    with open(FILE_PATH, 'w') as file:
        json.dump(data, file, indent=4)

# Initialize expenses data
expenses = load_data()

# Function to add an expense
def add_expense(amount, description, category):
    """Add a new expense entry to the list and save it."""
    expense = {
        'amount': amount,
        'description': description,
        'category': category,
        'date': datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    }
    expenses.append(expense)
    save_data(expenses)

# Function to get monthly summary
def get_monthly_summary():
    """Calculate and print a summary of expenses for the current month."""
    summary = {}
    current_month = datetime.now().strftime('%Y-%m')
    for expense in expenses:
        if expense['date'].startswith(current_month):
            category = expense['category']
            if category not in summary:
                summary[category] = 0
            summary[category] += expense['amount']
    return summary

# Function to search for expenses
def search_expenses(keyword):
    """Search for expenses by keyword in the description."""
    results = [expense for expense in expenses if keyword.lower() in
expense['description'].lower()]
    return results

# Function to edit an expense
def edit_expense(index, amount=None, description=None, category=None):
    """Edit an existing expense."""
    if index < 0 or index >= len(expenses):
        print("Invalid index.")
        return
    if amount:
        expenses[index]['amount'] = amount
    if description:
        expenses[index]['description'] = description
    if category:
        expenses[index]['category'] = category
    save_data(expenses)

# Function to delete an expense
def delete_expense(index):
    """Delete an existing expense."""
    if index < 0 or index >= len(expenses):
        print("Invalid index.")
        return
    del expenses[index]
    save_data(expenses)

# Function to export data to CSV
def export_to_csv(filename):
    """Export expenses data to a CSV file."""
    with open(filename, 'w', newline=") as csvfile:
        fieldnames = ['amount', 'description', 'category', 'date']
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        writer.writeheader()
        for expense in expenses:
            writer.writerow(expense)
    print(f"Data exported to {filename} successfully.")
```

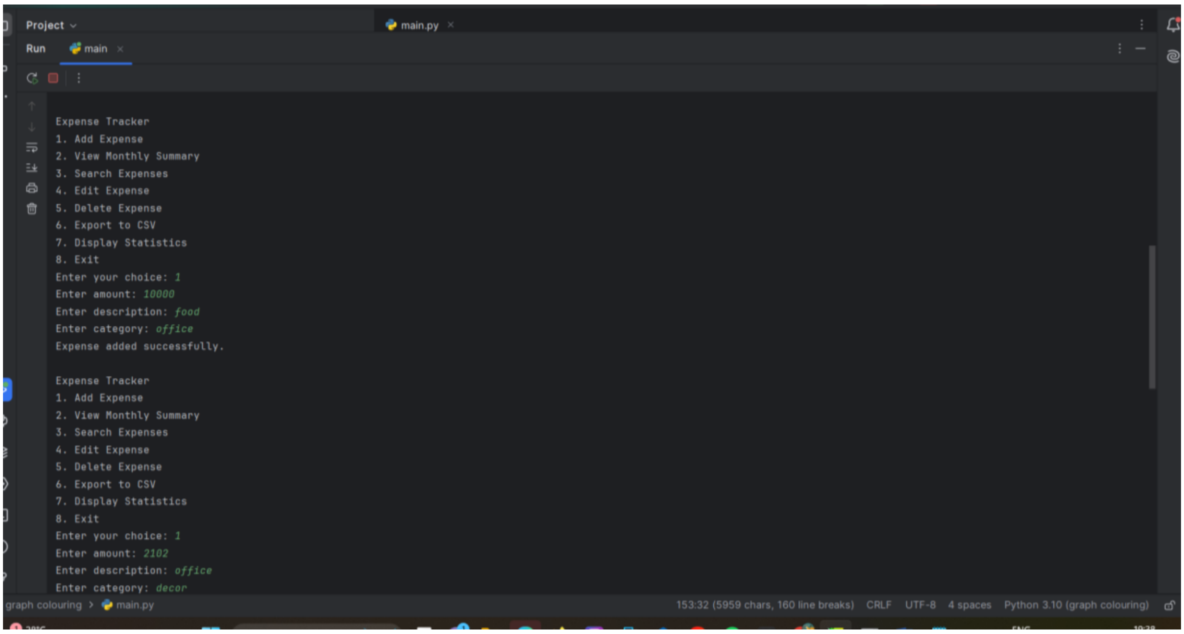
```
# Function to display statistics
def display_statistics():
    """Display basic statistics of expenses."""
    total_expenses = sum(expense['amount'] for expense in expenses)
    days_tracked = len(set(expense['date'][:10] for expense in expenses))
    average_per_day = total_expenses / days_tracked if days_tracked else 0
    print(f"Total Expenses: ${total_expenses:.2f}")
    print(f"Average Daily Spending: ${average_per_day:.2f}")

# User interface for interacting with the expense tracker
def user_interface():
    """Simple command-line interface for the expense tracker."""
    while True:
        print("\nExpense Tracker")
        print("1. Add Expense")
        print("2. View Monthly Summary")
        print("3. Search Expenses")
        print("4. Edit Expense")
        print("5. Delete Expense")
        print("6. Export to CSV")
        print("7. Display Statistics")
        print("8. Exit")
        choice = input("Enter your choice: ")

        if choice == '1':
            try:
                amount = float(input("Enter amount: "))
                description = input("Enter description: ")
                category = input("Enter category: ")
                add_expense(amount, description, category)
                print("Expense added successfully.")
            except ValueError:
                print("Invalid input. Please enter the correct values.")
        elif choice == '2':
            summary = get_monthly_summary()
            print("\nMonthly Summary:")
            for category, total in summary.items():
                print(f"{category}: ${total:.2f}")
        elif choice == '3':
            keyword = input("Enter a keyword to search: ")
            results = search_expenses(keyword)
            print("\nSearch Results:")
            for idx, expense in enumerate(results):
                print(f"{idx}. {expense['date']}: {expense['description']} - ${expense['amount']:.2f} ({expense['category']})")
        elif choice == '4':
            index = int(input("Enter the index of the expense to edit: "))
            amount = input("Enter new amount (leave blank to keep current): ")
            description = input("Enter new description (leave blank to keep current): ")
            category = input("Enter new category (leave blank to keep current): ")
            amount = float(amount) if amount else None
            edit_expense(index, amount, description, category)
            print("Expense edited successfully.")
        elif choice == '5':
            index = int(input("Enter the index of the expense to delete: "))
            delete_expense(index)
            print("Expense deleted successfully.")
        elif choice == '6':
            filename = input("Enter the filename to export to (with .csv extension): ")
            export_to_csv(filename)
        elif choice == '7':
            display_statistics()
        elif choice == '8':
            print("Exiting...")
            break
        else:
            print("Invalid choice. Please try again.")

# Run the user interface
if __name__ == '__main__':
    user_interface()
```

OUTPUT SCREENSHOT:

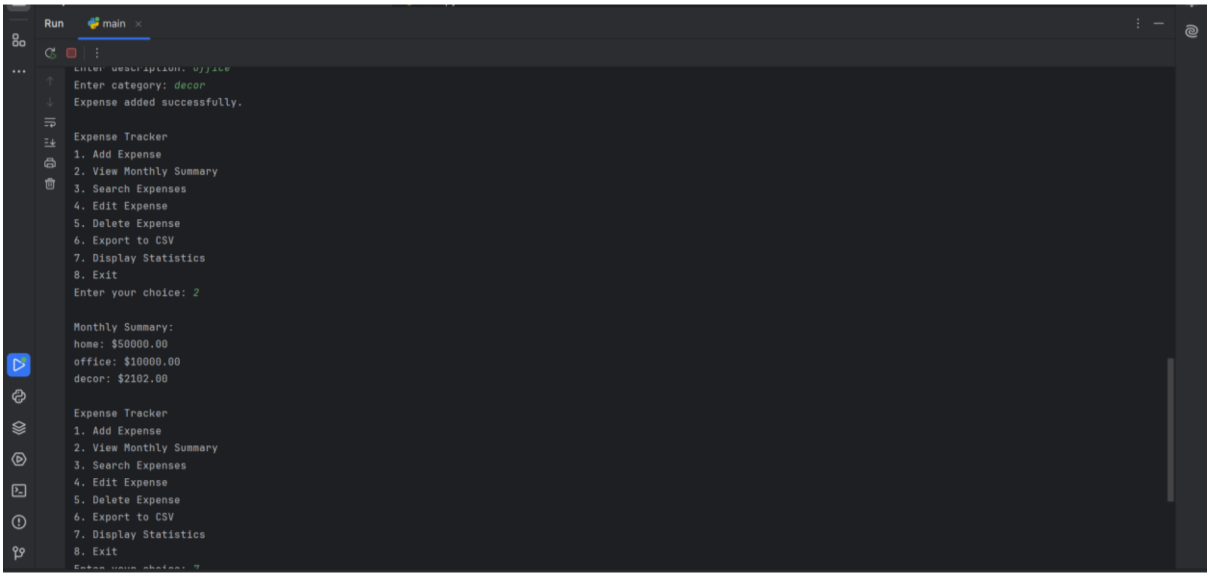


```
"C:\Users\pavan\PycharmProjects\26\06\graph colouring\.venv\Scripts\python.exe" "C:\Users\pavan\PycharmProjects\26\06\graph colouring\main.py"

Expense Tracker
1. Add Expense
2. View Monthly Summary
3. Search Expenses
4. Edit Expense
5. Delete Expense
6. Export to CSV
7. Display Statistics
8. Exit
Enter your choice: 1
Enter amount: 20000
Enter description: grocery
Enter category: home
Expense added successfully.

Expense Tracker
1. Add Expense
2. View Monthly Summary
3. Search Expenses
4. Edit Expense
5. Delete Expense
6. Export to CSV
7. Display Statistics
8. Exit
Enter your choice: 1
Enter amount: 30000
Enter description: car
```

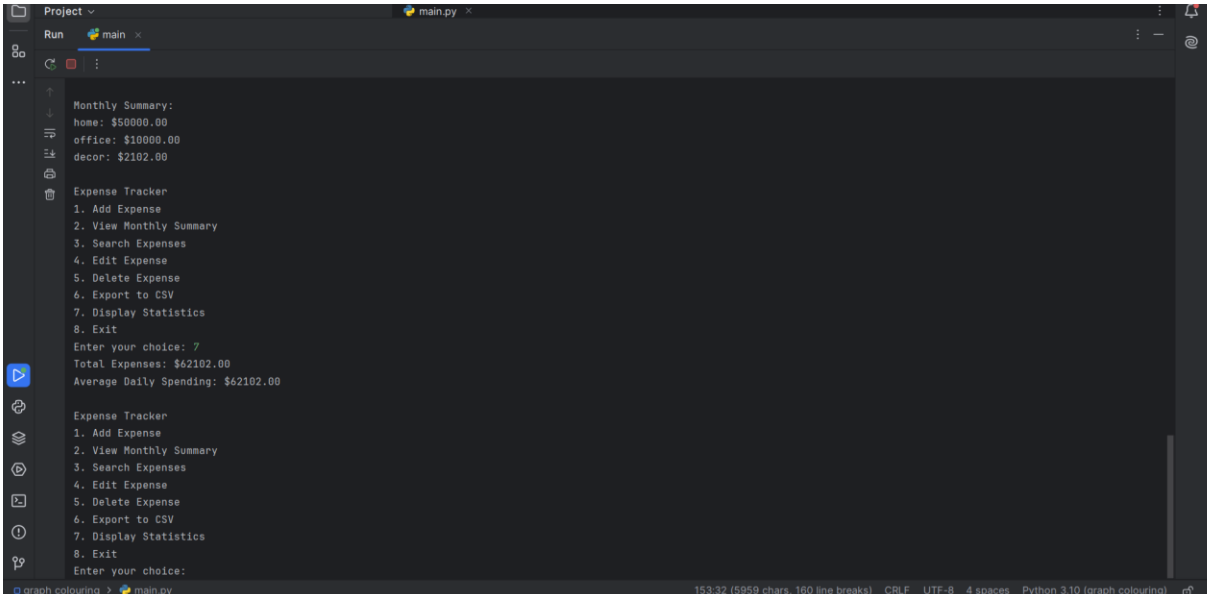
graph colouring > main.py
153:32 (5959 chars, 160 line breaks)
CRLF
UTF-8
4 spaces
Python 3.10 (graph colouring)


Project
main.py
Run
main
graph colouring > main.py
153:32 (5959 chars, 160 line breaks)
CRLF
UTF-8
4 spaces
Python 3.10 (graph colouring)

```
Expense Tracker
1. Add Expense
2. View Monthly Summary
3. Search Expenses
4. Edit Expense
5. Delete Expense
6. Export to CSV
7. Display Statistics
8. Exit
Enter your choice: 1
Enter amount: 10000
Enter description: food
Enter category: office
Expense added successfully.

Expense Tracker
1. Add Expense
2. View Monthly Summary
3. Search Expenses
4. Edit Expense
5. Delete Expense
6. Export to CSV
7. Display Statistics
8. Exit
Enter your choice: 1
Enter amount: 2102
Enter description: office
Enter category: decor
```



```
Enter description: office
Enter category: decor
Expense added successfully.

Expense Tracker
1. Add Expense
2. View Monthly Summary
3. Search Expenses
4. Edit Expense
5. Delete Expense
6. Export to CSV
7. Display Statistics
8. Exit
Enter your choice: 2

Monthly Summary:
home: $50000.00
office: $10000.00
decor: $2102.00

Expense Tracker
1. Add Expense
2. View Monthly Summary
3. Search Expenses
4. Edit Expense
5. Delete Expense
6. Export to CSV
7. Display Statistics
8. Exit
Enter your choice: 7
```



Project
main.py
Run
main
graph colouring > main.py
153:32 (5959 chars, 160 line breaks)
CRLF
UTF-8
4 spaces
Python 3.10 (graph colouring)

```
Monthly Summary:
home: $50000.00
office: $10000.00
decor: $2102.00

Expense Tracker
1. Add Expense
2. View Monthly Summary
3. Search Expenses
4. Edit Expense
5. Delete Expense
6. Export to CSV
7. Display Statistics
8. Exit
Enter your choice: 7
Total Expenses: $62102.00
Average Daily Spending: $62102.00

Expense Tracker
1. Add Expense
2. View Monthly Summary
3. Search Expenses
4. Edit Expense
5. Delete Expense
6. Export to CSV
7. Display Statistics
8. Exit
Enter your choice:
```