

```
smart-expense-manager.py > ...
1 import csv
2 import datetime
3
4 FILE_NAME = "expenses.csv"
5
6 # -----
7 # Function to add an expense
8 #
9 def add_expense():
10     date = input("Enter date (YYYY-MM-DD) or press enter for today: ")
11     if date == "":
12         date = datetime.date.today().isoformat()
13     category = input("Enter category (Food, Travel, Shopping, Bills etc.): ")
14     amount = float(input("Enter amount spent: "))
15
16     with open(FILE_NAME, "a", newline="") as file:
17         writer = csv.writer(file)
18         writer.writerow([date, category, amount])
19
20     print("Expense added successfully!\n")
21
22
23 # -----
24 # Function to view all expenses
25 #
26 def view_expenses():
27     try:
28         with open(FILE_NAME, "r") as file:
29             reader = csv.reader(file)
30             print("\n--- ALL EXPENSES ---")
31             for row in reader:
32                 print(f"Date: {row[0]}, Category: {row[1]}, Amount: ₹{row[2]}")
33             print()
34     except FileNotFoundError:
35         print("No expenses found! Add some first.\n")
36
```

Welcome

.py.txt 9+

smart expense manager.py X

import json Untitled-1 ●

D ⌘ ⌘ ⌘ ⌘ ⌘ ⌘ ⌘ ⌘ ⌘ ⌘

```
 26 def view_expenses():
 27     try:
 28         with open(FILE_NAME, "r") as file:
 29             reader = csv.reader(file)
 30             for row in reader:
 31                 print(row)
 32             total += float(row[2])
 33         print(f"\nTotal Expenses: ₹{total}\n")
 34     except FileNotFoundError:
 35         print("No expenses found!\n")
 36
 37
 38 # -----
 39 # Function to show total expenses
 40 #
 41 def total_expenses():
 42     try:
 43         total = 0
 44         with open(FILE_NAME, "r") as file:
 45             reader = csv.reader(file)
 46             for row in reader:
 47                 total += float(row[2])
 48         print(f"\nTotal Expenses: ₹{total}\n")
 49     except FileNotFoundError:
 50         print("No expenses found!\n")
 51
 52
 53 # -----
 54 # Function to show monthly expense summary
 55 #
 56 def monthly_summary():
 57     try:
 58         month = input("Enter month (MM): ")
 59         year = input("Enter year (YYYY): ")
 60
 61         total = 0
 62         print(f"\n--- Summary for {month}/{year} ---")
 63         with open(FILE_NAME, "r") as file:
 64             reader = csv.reader(file)
 65             for row in reader:
 66                 date = row[0]
 67                 if date.startswith(f"{year}-{month}"):
 68                     total += float(row[2])
 69             print(f"Date: {row[0]} Category: {row[1]} Amount: ₹{row[2]}\n")
 70
 71
 72 # -----
 73 # Function to add new expense
 74 #
 75 def add_expense():
 76     try:
 77         with open(FILE_NAME, "a") as file:
 78             writer = csv.writer(file)
 79             writer.writerow([date, category, amount])
 80             print("Expense added successfully!")
 81     except Exception as e:
 82         print(f"An error occurred: {e}")
 83
 84
 85 # -----
 86 # Function to update existing expense
 87 #
 88 def update_expense():
 89     try:
 90         with open(FILE_NAME, "r") as file:
 91             reader = csv.reader(file)
 92             rows = list(reader)
 93
 94             date = input("Enter date (DD/MM/YYYY): ")
 95             category = input("Enter category: ")
 96             amount = float(input("Enter amount: ₹"))
 97
 98             for row in rows:
 99                 if row[0] == date and row[1] == category:
 100                     row[2] = str(amount)
 101
 102             with open(FILE_NAME, "w") as file:
 103                 writer = csv.writer(file)
 104                 writer.writerows(rows)
 105                 print("Expense updated successfully!")
 106
 107
 108 # -----
 109 # Function to delete expense
 110 #
 111 def delete_expense():
 112     try:
 113         with open(FILE_NAME, "r") as file:
 114             reader = csv.reader(file)
 115             rows = list(reader)
 116
 117             date = input("Enter date (DD/MM/YYYY): ")
 118             category = input("Enter category: ")
 119
 120             for row in rows:
 121                 if row[0] == date and row[1] == category:
 122                     rows.remove(row)
 123
 124             with open(FILE_NAME, "w") as file:
 125                 writer = csv.writer(file)
 126                 writer.writerows(rows)
 127                 print("Expense deleted successfully!")
 128
 129
 130 # -----
 131 # Function to search expense by date and category
 132 #
 133 def search_expense():
 134     try:
 135         with open(FILE_NAME, "r") as file:
 136             reader = csv.reader(file)
 137             rows = list(reader)
 138
 139             date = input("Enter date (DD/MM/YYYY): ")
 140             category = input("Enter category: ")
 141
 142             for row in rows:
 143                 if row[0] == date and row[1] == category:
 144                     print(f"Date: {row[0]} Category: {row[1]} Amount: ₹{row[2]}\n")
 145
 146
 147 # -----
 148 # Main function to run the application
 149 #
 150 def main():
 151     while True:
 152         choice = input("Choose an option:\n1. View Expenses\n2. Total Expenses\n3. Monthly Summary\n4. Add Expense\n5. Update Expense\n6. Delete Expense\n7. Search Expense\n8. Exit\n")
 153
 154         if choice == "1":
 155             view_expenses()
 156         elif choice == "2":
 157             total_expenses()
 158         elif choice == "3":
 159             monthly_summary()
 160         elif choice == "4":
 161             add_expense()
 162         elif choice == "5":
 163             update_expense()
 164         elif choice == "6":
 165             delete_expense()
 166         elif choice == "7":
 167             search_expense()
 168         elif choice == "8":
 169             print("Exiting the application...")
 170             break
 171
 172
 173 # Run the main function
 174 main()
```

The screenshot shows a code editor interface with the following details:

- File Bar:** File, Edit, Selection, ...
- Search Bar:** Q smart-expense-manager
- Toolbar:** Back, Forward, Refresh, Home, Stop, Minimize, Maximize, Close.
- Left Panel:** Welcome, .py.txt 9+, smart expense manager.py (active), import json Untitled-1.
- Right Panel:** A vertical sidebar with multiple tabs and sections, likely a file browser or history.
- Code Area:** The main content is a Python script for a "smart expense manager". It includes functions for monthly summary and category-wise spending, both reading from a CSV file named FILE\_NAME.

```
def monthly_summary():
    for row in reader:
        date = row[0]
        if date.startswith(f"{year}-{month}"):
            total += float(row[2])
            print(f"Date: {row[0]}, Category: {row[1]}, Amount: ₹{row[2]}")
    print(f"\nTotal for {month}/{year}: ₹{total}\n")

except FileNotFoundError:
    print("No expenses found!\n")

# -----
# Function to view category-wise spending
# -----
def category_summary():
    try:
        category = input("Enter category to check: ")
        total = 0
        print(f"\n--- Category Summary: {category} ---")

        with open(FILE_NAME, "r") as file:
            reader = csv.reader(file)
            for row in reader:
                if row[1].lower() == category.lower():
                    print(f"Date: {row[0]}, Amount: ₹{row[2]}")
                    total += float(row[2])
            print(f"\nTotal spent on {category}: ₹{total}\n")

    except FileNotFoundError:
        print("No expenses found!\n")
```

Welcome

.py.txt 9+

smart expense manager.py X

import json Untitled-1 ●

▷ ▾

smart expense manager.py > ...

```
97     # -----
98     # Main Program Menu
99     # -----
100    while True:
101        print("===== SMART EXPENSE MANAGER =====")
102        print("1. Add Expense")
103        print("2. View All Expenses")
104        print("3. View Total Expenses")
105        print("4. Monthly Summary")
106        print("5. Category-wise Summary")
107        print("6. Exit")
108
109        choice = input("Enter your choice: ")
110
111        if choice == "1":
112            add_expense()
113        elif choice == "2":
114            view_expenses()
115        elif choice == "3":
116            total_expenses()
117        elif choice == "4":
118            monthly_summary()
119        elif choice == "5":
120            category_summary()
121        elif choice == "6":
122            print("Thank you for using Smart Expense Manager!")
123            break
124        else:
125            print("Invalid choice! Try again.\n")
```