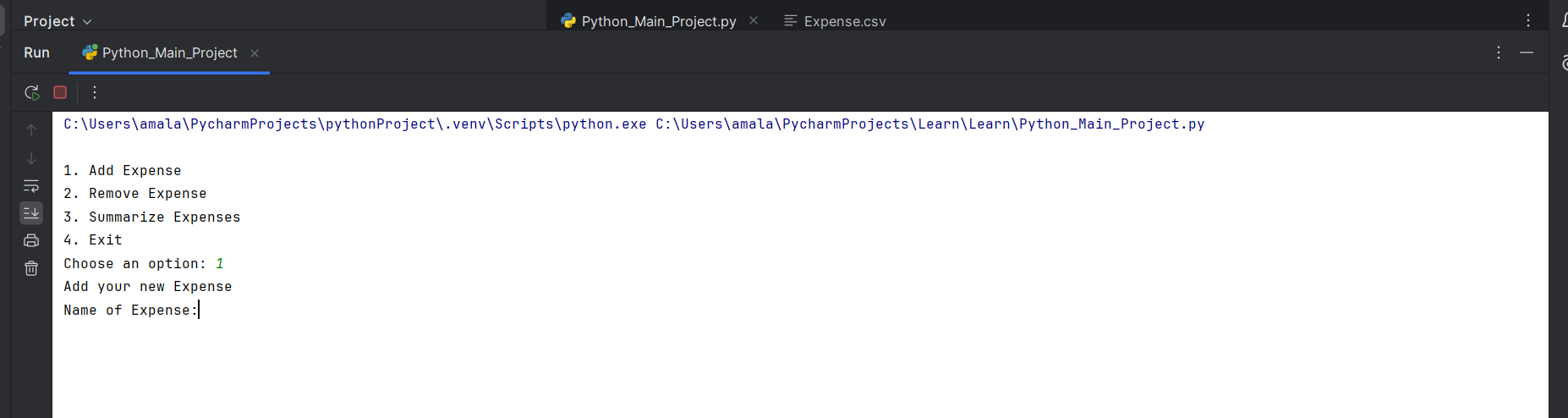
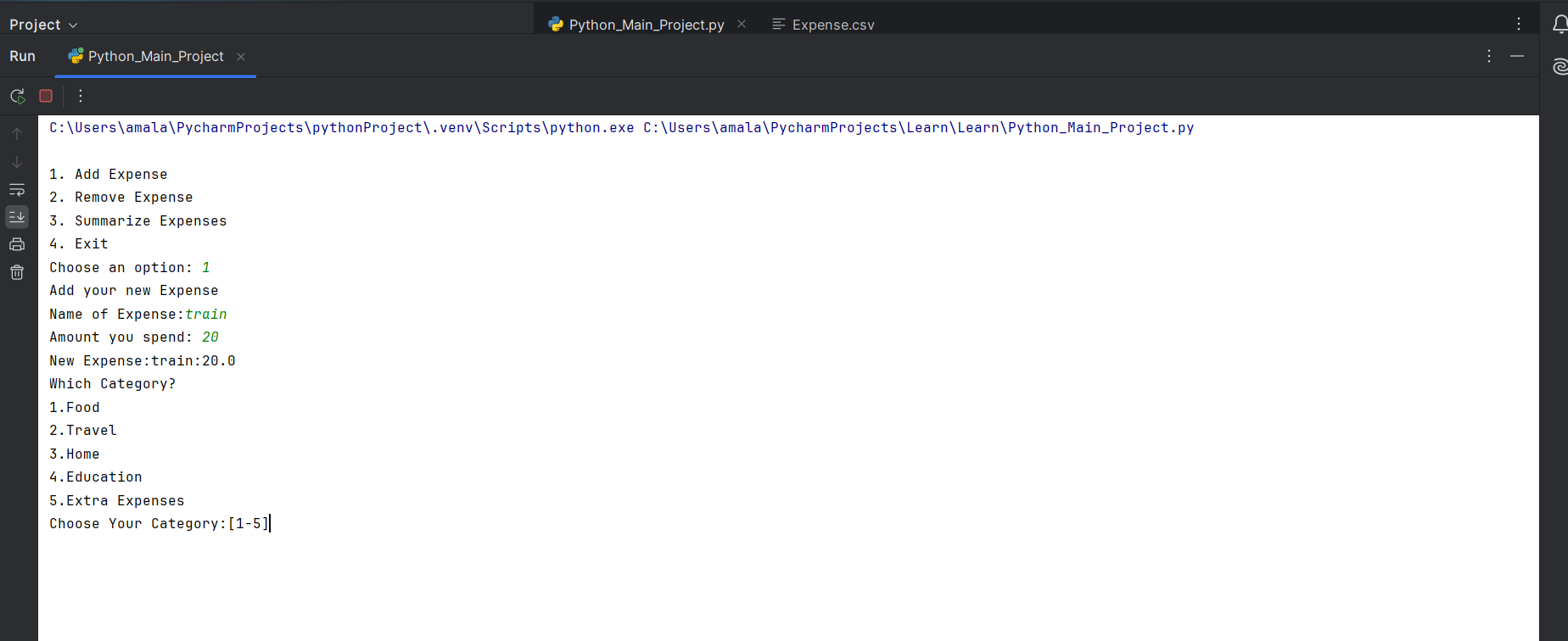
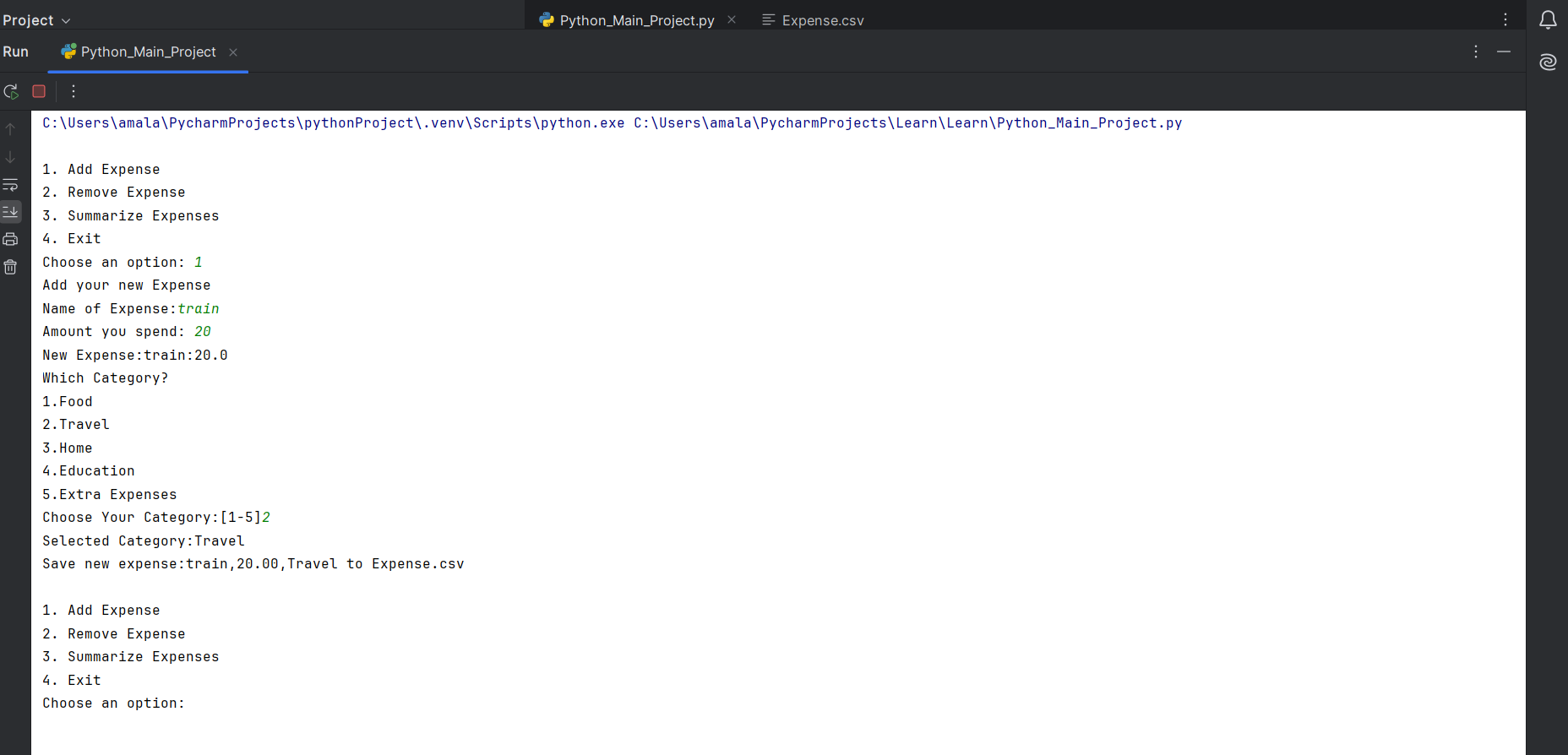
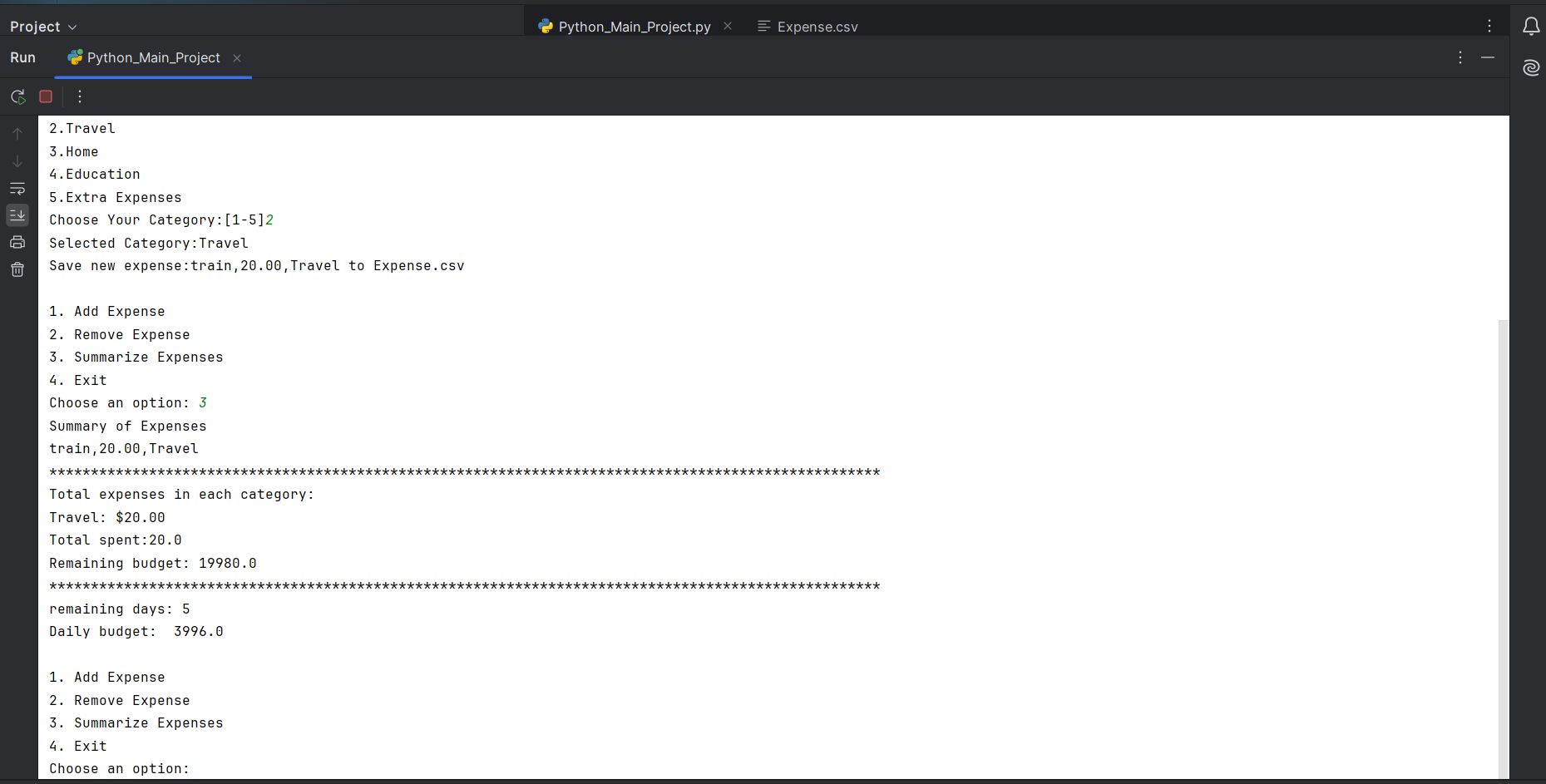
**EXPENSE TRACKER**

**OUTPUT**

****

****

****

****

**Input**

import datetime  
import calendar  
class Expense:  
 def \_\_init\_\_(self,name,amount,category):  
 self.name=name  
 self.amount=amount  
 self.category=category  
 def \_\_repr\_\_(self):  
 return f"{self.name},{self.amount:.2f},{self.category}"  
def expense\_details():  
 expense\_file\_path='Expense.csv'  
 budget=20000  
 while True:  
 print("\n1. Add Expense\n2. Remove Expense\n3. Summarize Expenses\n4. Exit")  
 choice = input("Choose an option: ")  
 if choice == '1':  
 add = add\_expense()  
 save\_file(add, expense\_file\_path)  
 elif choice == '2':  
 remove\_expense(expense\_file\_path)  
 elif choice == '3':  
 summarize\_expense(expense\_file\_path, budget)  
 elif choice == '4':  
 break  
 else:  
 print("Invalid choice. Please try again.")  
def add\_expense():  
 print('Add your new Expense')  
 new\_expense=input("Name of Expense:")  
 new\_amount=float(input('Amount you spend: '))  
 print(f"New Expense:{new\_expense}:{new\_amount}")  
 expense\_categories=['Food','Travel','Home','Education','Extra Expenses']  
 while True:  
 print('Which Category?')  
 for i,category\_name in enumerate(expense\_categories):  
 print(f"{i+1}.{category\_name}")  
 value\_range=f"[1-{len(expense\_categories)}]"  
 selected\_index=int(input(f"Choose Your Category:{value\_range}"))-1  
 if selected\_index in range(len(expense\_categories)):  
 selected\_category=expense\_categories[selected\_index]  
 print(f"Selected Category:{selected\_category}")  
 expense=Expense(name=new\_expense,amount=new\_amount,category=selected\_category)  
 return expense  
 else:  
 print('invalid syntax')  
 break  
def remove\_expense(expense\_file\_path):  
 expense\_name\_to\_remove = input("Enter the name of the expense to remove: ").strip()  
 updated\_expenses = []  
 with open(expense\_file\_path, 'r') as f:  
 lines = f.readlines()  
 for line in lines:  
 line = line.strip()  
 if line:  
 parts = line.split(',')  
 if len(parts) == 3:  
 expense\_name, expense\_amount, expense\_category = parts  
 if expense\_name != expense\_name\_to\_remove:  
 updated\_expenses.append(line)  
 else:  
 print(f"Removing expense: {line}")  
 else:  
 print(f"Skipping improperly formatted line: {line}")  
  
 with open(expense\_file\_path, 'w') as f:  
 for expense in updated\_expenses:  
 f.write(f"{expense}\n")  
 print(f"Expense {expense\_name\_to\_remove} removed successfully.")  
def save\_file(add:Expense,expense\_file\_path):  
 print(f"Save new expense:{add} to {expense\_file\_path}")  
 with open(expense\_file\_path,"a") as f:  
 f.write(f"{add.name},{add.amount:.2f},{add.category}\n")  
 pass  
def summarize\_expense(expense\_file\_path,budget):  
 print(f"Summary of Expenses")  
 list=[]  
 with open(expense\_file\_path, 'r') as f:  
 lines = f.readlines()  
 for line in lines:  
 line = line.strip()  
 if line:  
 parts = line.split(',')  
 if len(parts) == 3:  
 expense\_name, expense\_amount, expense\_category = parts  
 print(f"{expense\_name},{expense\_amount},{expense\_category}")  
 list.append(Expense(name=expense\_name,amount=float(expense\_amount),category=expense\_category))  
 else:  
 print(f"Skipping improperly formatted line: {line}")  
 amount\_by\_category = {}  
 for expense in list:  
 key = expense.category  
 if key in amount\_by\_category:  
 amount\_by\_category[key] += expense.amount  
 else:  
 amount\_by\_category[key] = expense.amount  
 print('\*' \* 100)  
 print("Total expenses in each category:")  
 for key, amount in amount\_by\_category.items():  
 print(f"{key}: ${amount:.2f}")  
 total\_spent=sum([x.amount for x in list])  
 print(f"Total spent:{total\_spent}")  
 remaining\_budget=budget-total\_spent  
 print(f"Remaining budget:",remaining\_budget)  
 now=datetime.datetime.now()  
 days\_in\_month=calendar.monthrange(now.year,now.month)[1]  
 print('\*'\*100)  
 remaining\_days=days\_in\_month-now.day  
 print(f"remaining days:",remaining\_days)  
 daily\_budget=remaining\_budget/remaining\_days  
 print(f"Daily budget: ",daily\_budget)  
expense\_details()