Mini Project

PROGRAM:

```
Import csv
From datetime import datetime

# Define the filename for storing expenses
FILENAME = ‘expenses.csv’

Def load_expenses():
  “”””Load expenses from a CSV file.”””
  Expenses = []
  Try:
    With open(FILENAME, mode=’r’, newline=’’) as file:
      Reader = csv.DictReader(file)
      For row in reader:
        Row[‘Amount’] = float(row[‘Amount’])
        Row[‘Date’] = datetime.strptime(row[‘Date’], ‘%Y-%m-%d’)
        Expenses.append(row)
  Except FileNotFoundError:
    Pass  # File doesn’t exist yet
  Return expenses

Def save_expenses(expenses):
  “”””Save expenses to a CSV file.”””
```

```
    With open(FILENAME, mode='w', newline='') as file:
        Fieldnames = ['Amount', 'Category', 'Date']
        Writer = csv.DictWriter(file, fieldnames=fieldnames)
        Writer.writeheader()
        For expense in expenses:
            Writer.writerow(expense)

Def add_expense(expenses):
    """Add a new expense to the list."""
    Amount = float(input("Enter the amount: "))
    Category = input("Enter the category: ")
    Date_input = input("Enter the date (YYYY-MM-DD) or press Enter for today: ")
    Date = datetime.strptime(date_input, '%Y-%m-%d') if date_input else datetime.now()

    Expense = {
        'Amount': amount,
        'Category': category,
        'Date': date.strftime('%Y-%m-%d')
    }
    Expenses.append(expense)
    Save_expenses(expenses)
    Print("Expense added successfully!")

Def view_summary(expenses):
    """View the summary of expenses."""
    Total_spending = sum(expense['Amount'] for expense in expenses)
```

```
    Print(f"\nTotal spending: ${total_spending:.2f}")

    Category_summary = {}
    For expense in expenses:
        Category = expense['Category']
        Category_summary[category] = category_summary.get(category, 0) + expense['Amount']

    Print("\nSpending by category:")
    For category, total in category_summary.items():
        Print(f"{category}: ${total:.2f}")

Def main():
    """Main function to run the expense tracker."""
    Expenses = load_expenses()

    While True:
        Print("\nPersonal Expense Tracker")
        Print("1. Add Expense")
        Print("2. View Summary")
        Print("3. Exit")

        Choice = input("Choose an option: ")
        If choice == '1':
            Add_expense(expenses)
        Elif choice == '2':
            View_summary(expenses)
```

```
    Elif choice == '3':
        Print("Exiting the program.")
        Break
    Else:
        Print("Invalid choice. Please try again.")

If __name__ == "__main__":
  Main()
```

OUTPUT:

```
Personal Expense Tracker
1. Add Expense
2. View Summary
3. Exit
Choose an option: 1
Enter the amount: 2000
Enter the category: Shopping
Enter the date (YYYY-MM-DD) or press Enter for today: 2024-10-15
Expense added successfully!

Personal Expense Tracker
1. Add Expense
2. View Summary
3. Exit
Choose an option: 2
```

Total spending: $2000.00

Spending by category:

Shopping: $2000.00

Personal Expense Tracker

1. Add Expense

2. View Summary

3. Exit

Choose an option: 3

Exiting the program.

EXPLANATION:

Starting Up: When you run the program, it looks for a file called expenses.csv where your expenses are stored. If it finds this file, it loads the data into memory so you can see your past expenses.

Adding Expenses: You can choose to add a new expense. The program will ask you for the amount you spent, what category it falls into (like groceries or entertainment), and the date of the expense. If you don't specify a date, it will automatically use today's date. Once you enter this information, it saves the new expense to the CSV file so you don't lose it.

Viewing Your Spending: If you want to see how much you've spent in total or get a breakdown by category, you can select the summary option. The program calculates your total spending and organizes it by category, showing you how much you've spent in each area. This helps you see where your money is going at a glance.

User Interaction: The whole process is interactive. You're presented with a simple menu where you can choose to add an expense, view your spending summary, or exit the program. If you enter something that isn't an option, it gently prompts you to try again.

Closing the Program: When you're done, you can exit the program, and your expenses will still be saved in the CSV file for next time.