

# Python Developer Internship Project: Personal Expense Tracker

## Project Title: Personal Expense Tracker

Objective:

- Develop a Python-based application that allows users to track their personal expenses by recording, categorizing, and analyzing their spending patterns.

## Phase 1: Understanding Requirements and Basic Design

Key Points:

- Define the main features:
  - Add expenses with details (amount, date, category, description).
  - View expenses.
  - Generate basic reports of spending.
- Design the program flow:
  - Use a command-line interface to interact with the user.
  - Outline the basic functions: `add_expense()`, `view_expenses()`, `save_expenses()`, and `load_expenses()`.

## Phase 2: Implement Core Functionality

Objective: Develop the core functionality of the application.

Tasks and Code:

1. Adding an Expense:

# Python Developer Internship Project: Personal Expense Tracker

```
expenses = [] # List to store expense entries

def add_expense():

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

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

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

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

    expense = {

        'amount': amount,

        'date': date,

        'category': category,

        'description': description

    }

    expenses.append(expense)

    print("Expense added successfully!\n")
```

## 2. Viewing Expenses:

```
def view_expenses():

    if not expenses:

        print("No expenses recorded yet.")

    return
```

# Python Developer Internship Project: Personal Expense Tracker

```
print("\nAll Recorded Expenses:")

for i, expense in enumerate(expenses, 1):

    print(f"{i}. Amount: {expense['amount']}, Date: {expense['date']}, "
          f"Category: {expense['category']}, Description: {expense['description']}")

print()
```

## Phase 3: Data Storage and Retrieval

Objective: Save expenses to a file and load them when the program starts.

### 1. Saving Expenses to a File (CSV format):

```
import csv

def save_expenses():

    with open('expenses.csv', mode='w', newline='') as file:

        writer = csv.writer(file)

        writer.writerow(['Amount', 'Date', 'Category', 'Description'])

        for expense in expenses:

            writer.writerow([expense['amount'], expense['date'],
                             expense['category'], expense['description']])

    print("Expenses saved to file.\n")
```

# Python Developer Internship Project: Personal Expense Tracker

## Phase 4: Data Analysis and Visualization

Objective: Add basic data analysis and visualization using Matplotlib.

### 1. Analyze Expenses by Category:

```
from collections import defaultdict

def analyze_expenses():

    category_totals = defaultdict(float)

    for expense in expenses:

        category_totals[expense['category']] += expense['amount']

    print("\nExpense Summary by Category:")

    for category, total in category_totals.items():

        print(f"{category}: {total}")

    print()
```

## Phase 5: Testing and Debugging

Objective: Ensure the application functions correctly.

- Testing: Test each function individually to ensure correct outputs.

# Python Developer Internship Project: Personal Expense Tracker

- Debugging: Handle potential errors such as incorrect input formats or file read/write issues.

## Putting It All Together: Main Program

```
def main():  
  
    load_expenses()  
  
    while True:  
  
        print("Expense Tracker Menu:")  
  
        print("1. Add Expense")  
  
        print("2. View Expenses")  
  
        print("3. Save Expenses")  
  
        print("4. Analyze Expenses")  
  
        print("5. Visualize Expenses")  
  
        print("6. Exit")  
  
  
        choice = input("Enter your choice: ")  
  
  
        if choice == '1':  
  
            add_expense()  
  
        elif choice == '2':  
  
            view_expenses()  
  
        elif choice == '3':  
  
            save_expenses()  
  
        elif choice == '4':
```

## Python Developer Internship Project: Personal Expense Tracker

```
        analyze_expenses()

elif choice == '5':

    visualize_expenses()

elif choice == '6':

    print("Exiting the application. Goodbye!")

    break

else:

    print("Invalid choice. Please try again.\n")

if __name__ == "__main__":

    main()
```