VITYARTHI – PROJECT

uu

Project title : Personal Expense Tracker

Course title : Introduction to problem solving and programming

Course code : CSE 1021

Course type : Flipped course

Course credits : 4

Professor : Dr. G.Prabhu Kanna

Slot : C11+C12+C13

Submitted by : Bavesh.P

Registration number : 25BAI10093

Submission date : 23/11/25

VITYARTHI – PROJECT

INTRODUCTION :

Personal Expense Tracker is made to provide you with a clear, fair view of your spending day to day habits. By entering every transaction, you will effortlessly transform unclear financial worries into specific, actionable data.

Personal expense tracker is more than just a file of accounts; it's your daily co-pilot on the journey to financial freedom. We believe that mindful spending is the foundation of a successful financial life by making the simple commitment to track your expenses.

This personal expense tracker is a tool or a system designed to help individuals monitor, record, and analyse their daily spending and income to better manage their personal finances.

PROBLEM STATEMENT :

The core problem a personal expense tracker addresses is the difficulty many individuals face in understanding and controlling their spending habits using traditional manual methods without this personal expense tracker we can't able look or track our daily expenses and the complexity in maintaining the paper-based expense tracker increases beyond the time complexity so I have created this "Personal expense tracker" as the solution for this trouble

VITYARTHI – PROJECT

PROJECT OBJECTIVES :

- Creating a personal expense tracker provide users with a clear and centralised record of all income and expenditure
- It is used to enable users to set spending limits, track their progress against those limits in real time
- Tracking for active working towards long term financial objectives (e.g. saving for medical usage, paying loan)
- It helps as supporter in major financial goal achievement
- It gives a great plan to secure data storage handling practices to protect user financial information

FUNCTIONAL REQUIREMENTS:

1.Expense management:

- Add New Expenses : This code allows the user to input and store details for a new expense
  Details required: item name
- View all expense : The system must display a list of all currently recorded expenses
- Calculate total expenses : The system must be able to sum the amounts of all recorded expenses and display the total.

VITYARTHI – PROJECT

2.User Interface and Navigation :

- Display main menu :The code must present a clear navigable text -based menu
- Process user choice: The code must accept user input
- Handle invalid input : The system must recognise and handle invalid menu choices
- Exit program : The code must provide a defined mechanism to terminate the program.

3.Data Handling:

- Store expenses : The system must use a list to hold expense record stored as a dictionary

NON-FUNCTIONAL REQUIREMENTS :

1.Performance:

- The system should response to user actions within 1 seconds
- The application should load and present the main menu within 2 seconds

2.Security:

- Since the tracker deals with sensitive personal financial data, the data stored must be protected from unauthorised access

3.Usability:

- The user interface must be intuitive and easy to understand
- The process for adding a new expense should be quick and require minimal number of inputs

4.Reliability:

- The system must accurately calculate and report the totals
- The tracker should be available for use whenever the user launches the application
- The data entered should not be lost during operation

5.Maintanability:

- The code base should be well organised and use clear function names and comments to allow a developer to easily understand, debug and modify the code

6. Error handling :
    - If the system detects any error in the inputs entered by the user it displays an error message and requests' the user to try entering the inputs again.

7.Resource efficiency :
    - The system is designed to minimise power consumption and optimize battery life.
    - The system shall require less than 10MB of storage space for optimised running.
    - The system doesn't require any network bandwidth so it can be accessed even in places without proper internet connection.

SYSTEM ARCHITECTURE :

- Presentation layer is built using Python IDLE where the inputs are accepted from the user.
- Application layer is where the input data is processed and the calculations are performed then the outputs are stored and displayed later.
- Data layer is a storage space where the user inputs and the output values are stored.
- Output renderer is a dynamic area that shows all results including detailed personal expenses on each day and alerts on errors.

VITYARTHI – PROJECT

DESIGN DIAGRAM :

1. Workflow :
   - Start ⟶ Display menu ⟶ Input choice ⟶ Decision points ⟶ End

2. Data flow Diagram :
   - Data store(Ds):
     The in-memory list where all transaction dictionaries are stored
   - Process1(P1) :
     Takes item name and input from the user and write it into the Ds.
   - Process2(P2):
     Reads from D1 and outputs the formatted list to the user.
   - Process3(P3):
     Reads all amounts from D1 and outputs the total to the user.
   - Process4(P4):
     Handles the user input and direct flow to P1,P2,P3 or exits

3. Component diagram :
   - Input form : Collects expense details from the user
   - Validation module : Checks all input ranges and types
   - Calculation core : It goes through every single expense entered in the list
   - Output display : Presents the results

VITYARTHI – PROJECT

DESIGN DECISIONS & RATIONALE :

1. Programming language : Python
- Rationale : Python is a simple, easy to learn language with large number of libraries and frameworks, making it idle for small projects like this.

2. User interface ; Python IDLE
- Rationale : A simple UI is sufficient for this program so IDLE is chosen which provides an interactive environment for user to input data and view results.

3. Data storage : In-memory data storage
- Rationale : Since the project doesn't require any explicit data storage, in-memory storage is sufficient and efficient.

4. Calculation Algorithm : Simple formula-based calculation
- Rationale : efficient way to aggregate values from a list of dictionaries

5. Error handling :
- Rationale : Basic error handling is sufficient for the small project like this and the user is expected to enter valid data.

IMPLEMENTATION DETAILS :

1. Language : Python
2. Development environment : Python IDLE
3. Module : IPSP_Project_Personal_expense_tracker

VITYARTHI – PROJECT

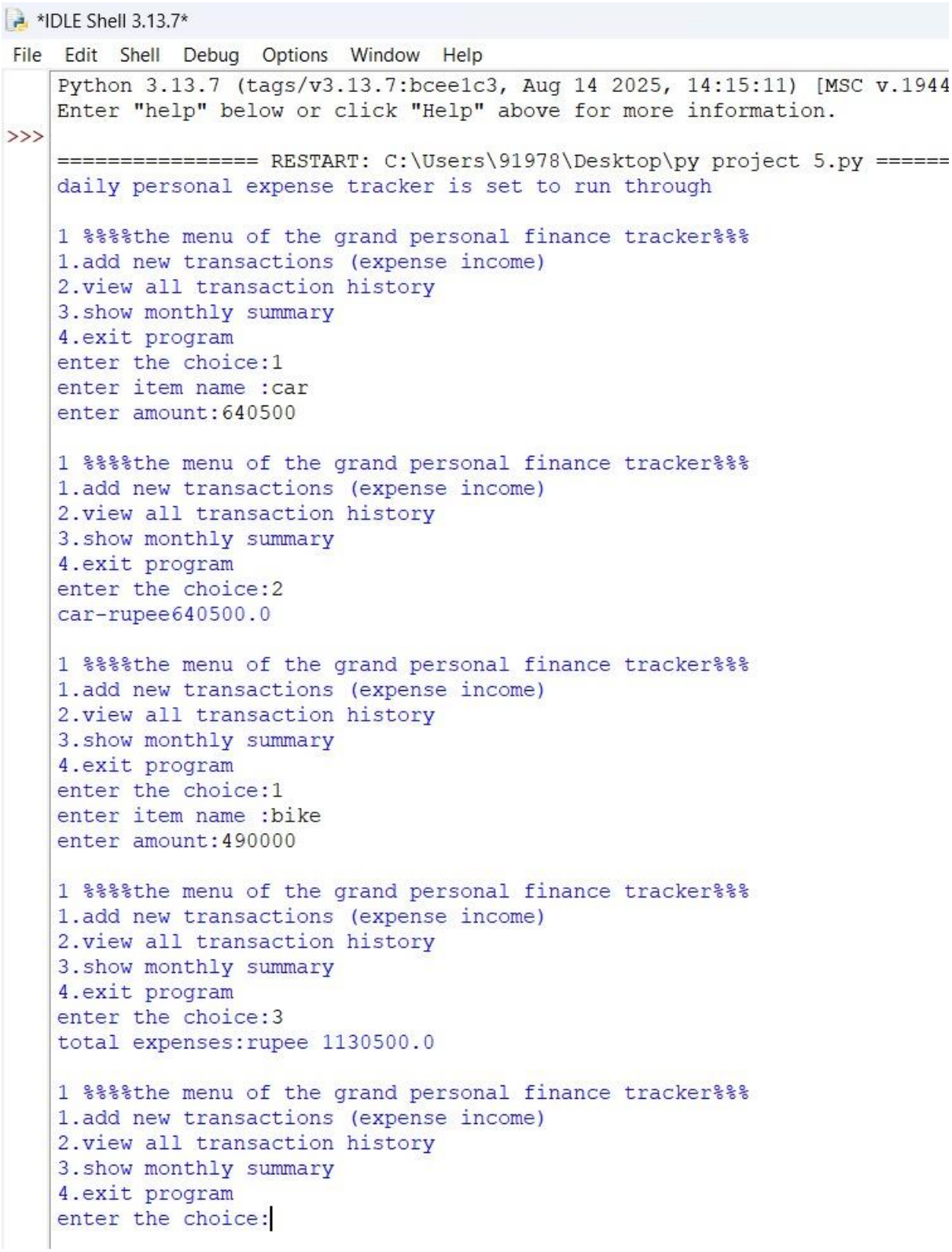
4. Implementation steps :
    1. Open a text editor
    2. Paste the code
    3. Save the file

TESTING APPROACH :

1. Unit tests :
   Manually testes and isolates each function to ensure it works correctly under various conditions
2. System Integration :
   System testing checks the application against the stated and implied user requirements .
3. Validation :
   Ensures the user inputs a valid number for the expense amount
4. User flow testing :
   Ensure that an error message display if any of the input it wrong.
5. Peer view :
   Verified among my peer group on calculations and reliability.

VITYARTHI – PROJECT

SCREENSHOTS / RESULTS :



VITYARTHI – PROJECT

CHALLENGES FACED :

1. Data integrity :
   The biggest challenge is the complete loss of all data every time the program exists.the expenses list is stored only in the computer's memory.
2. User error handling :
   The code will crash with a ValueError if the user enters non numeric text when entered the expense amount
3. Transparency :
   Providing the user with the clear step by step Calculations and not just their final results.

LEARNING AND KEY TAKEAWAYS :

1. Implementation :
   Learnt how to implement the acquired knowledge to solve the real-world problems, acquired a lot of practical skills in shape-shifting the academics into a key for locks of problem in today's world.

2. Automation value :
   This would have took a lot of time and energy if it was done manually but now within seconds with the help of this calculator we can get the results automatically in a few seconds.

3. Testing :
   Coding part is easy if we know the logic behind it but we can only know our code is correct when it test run it with multiple

values, so I understood that testing a system is as important as designing a system.

4. Communication :
   The system's ability of guidance has been improved and error messages are directly displayed in the output UI for optimal user support.

FUTURE ENHANCEMENTS :

1. Data persistence :
   Add a support for local storage, so that the user can store his previous session data and prevent it from losing in the next session.

2. User Interface :
   Separately designing a new User Interface (App / Webpage) for better conditions and presentation and instant access of the system rather than running it in Python IDLE

3. Analytics :
   With the help of the previous data entered by the user predicting their next most probable Investment and choice and creating graphical dashboards for easier access.

4. Export :
   Enabling exporting reports as in the form of PDF or EXCEL or Direct email share formats for the daily use for everyone's personal expense.

VITYARTHI – PROJECT

REFERENCE :

1. VITYARTHI and CSE1021 course for Python coding skills.
2. Google for learning about Documentation.
3. Peer and mentor feedbacks from,
    - Dr. G. Prabhu Kanna
    - Gokul – 22BCG10045,
    - Gopi – 23BSA10162,
    - Gokul Prasad. K - 25BAI10148,
    - Rahul – 25BAI10288
    - Sudharshan. SG – 25BAI10706,
    - Dilip. S – 25BAI10590.