

Project Report

on

Python Projects: Expense Tracker & Notes Manager

Submitted

In Partial Fulfillment of

MASTER OF COMPUTER APPLICATIONS (MCA)

Submitted by:

Pratyansh Sharma

Roll No: 24/SCA/BCA(AI&ML)/052

Under the Supervision of:

(Ms Iram Fatima)

School of Computer Applications

Manav Rachna International Institute of Research and Studies

Faridabad – 121001

September 2025

Declaration

I do hereby declare that this project work entitled “Python Projects: Expense Tracker & Notes Manager” submitted by me for the partial fulfillment of the requirement for the award of BACHELOR OF COMPUTER APPLICATIONS is a record of my own work. The report embodies the findings based on my study and observation and has not been submitted earlier for the award of any degree or diploma to any Institute or University.

SIGNATURE

Name: Pratyansh Sharma

Roll No: 24/SCA/BCA(AIML)/052

Date: 3 SEPTEMBER 2025

Certificate from the Guide

This is to certify that the project report entitled “Python Projects: Expense Tracker & Notes Manager” submitted in partial fulfillment of the degree of BACHELOR OF COMPUTER APPLICATIONS to Manav Rachna International Institute of Research and Studies, Faridabad is carried out by Pratyansh Sharma(Roll No 24/SCA/BCA(AIML)/052) under my guidance.

Signature of the Guide

Name: Ms Iram Fatima

Date: 3 SEPTEMBER 2025

Head of Department

Name: Dr. Sohail Javed

Date: 3 SEPTEMBER 2025

Acknowledgement

I gratefully acknowledge for the assistance, cooperation, guidance and clarification provided by my guide during the development of the projects. My extreme gratitude to all faculty members who guided and supported me throughout the project. Without their timely clarification, encouragement and support, this project could not have been completed in due time.

This opportunity is a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement to attain desired career objectives.

Index

I Cover Page	1
II Declaration	2
III Certificate	3
IV Acknowledgement	4
V Index	5
VI Introduction	6
VII System Study	8
VIII Feasibility Study	10
IX Project Monitoring (Gantt Chart)	12
X System Analysis	13
XI System Design	16
XII Input/Output Design	18
XIII System Testing	21
XIV System Implementation	23
XV Documentation	24
XVI Scope of the Project	26
XVII Bibliography	28

Introduction

This report presents two Python-based mini-projects developed as part of the BCA curriculum: 1) Expense Tracker and 2) Notes Manager. Both applications are designed with simplicity, practicality, and educational value in mind.

The Expense Tracker helps users record, categorize, and analyze their daily expenses. It demonstrates file handling, data persistence using JSON, and menu-driven programming in Python.

The Notes Manager allows users to create, search, edit, and delete notes, also using JSON for storage. It demonstrates CRUD operations (Create, Read, Update, Delete), search functionality, and modular code organization.

Together, these projects represent practical applications of Python for personal productivity and learning.

System Study

a) Existing System and Limitations

Currently, most students or individuals rely on either manual tracking of expenses and notes or use advanced, paid tools and mobile apps. These apps often require online accounts, data synchronization, and advertisements, which can be distracting for academic learning. Beginners find it difficult to understand how data is stored and retrieved in such systems.

Limitations of existing systems:

- Dependence on internet-based services.
- Complex user interfaces that are not beginner-friendly.
- Lack of transparency in how data is managed.
- Privacy concerns due to cloud storage.

b) Proposed System and Advantages

The proposed projects solve these issues by providing simple, offline, local-storage-based applications using Python.

Advantages:

- Beginner-friendly, simple text-based interface.
- Offline functionality using JSON files.
- Demonstrates practical Python concepts like file handling, loops, and conditionals.
- Zero cost, as it uses open-source tools.

Feasibility Study

a) Technical Feasibility

The projects use Python (an open-source language) and JSON files for storage. They run on any machine with Python installed, requiring no additional hardware or paid software.

b) Behavioural Feasibility

The applications are designed to be user-friendly, with clear menus and prompts. Any student or beginner can use them without prior training.

c) Economic Feasibility

The system has zero financial cost. Python is free, JSON is lightweight, and no internet or paid services are required.

Project Monitoring (Gantt Chart)

Task	Start Date	End Date	Duration
Planning	01/07/25	02/07/25	2 days
Learning & Setup	03/07/25	04/07/25	2 days
Expense Tracker Dev	05/07/25	08/07/25	4 days
Notes Manager Dev	09/07/25	12/07/25	4 days
Testing & Report	13/07/25	15/07/25	3 days

System Analysis

Hardware Requirements:

- Processor: Intel i3 or equivalent
- RAM: Minimum 2 GB
- Storage: 200 MB free space

Software Requirements:

- Python 3.8 or higher
- Any text editor or IDE (e.g., VS Code, PyCharm)
- JSON (built-in with Python)

Flowcharts:

Expense Tracker Flow:

User selects option → Add/View/Summary → Data stored/retrieved in JSON → Output shown to user.

Notes Manager Flow:

User selects option → Add/Search/Edit/Delete → Data stored/retrieved in JSON → Output displayed.

System Design

Expense Tracker Files:

- expenses.json – stores data
- main.py – contains logic for menu, input, output, and JSON handling

Notes Manager Files:

- notes.json – stores notes
- notes.py – contains functions for add, view, edit, delete

Both systems use modular functions for clarity and reusability.

Input/Output Design

Expense Tracker Input:

- Category, amount, date

Expense Tracker Output:

- Expense list, monthly summary, category-wise display

Notes Manager Input:

- Title, content, ID

Notes Manager Output:

- List of notes, search results, edited/deleted confirmation

System Testing

Test Cases:

Expense Tracker:

- Add Expense → Saved correctly → Pass
- Monthly Summary → Correct total → Pass

Notes Manager:

- Add Note → Appears in list → Pass
- Edit Note → Changes saved → Pass
- Delete Note → Removed successfully → Pass

System Implementation

How to Run:

1. Open terminal.
2. Run `python main.py` for Expense Tracker.
3. Run `python notes.py` for Notes Manager.

The applications run in a menu-driven format and store data in JSON files automatically.

Documentation

Expense Tracker:

- **Functionality:** Tracks expenses by category/date.
- **Technology:** Python, JSON.
- **Key Concepts:** File handling, data storage, menu-driven programming.

Notes Manager:

- **Functionality:** Manages notes with CRUD operations.
- **Technology:** Python, JSON.
- **Key Concepts:** CRUD operations, searching, modular functions.

Scope of the Project

In Scope:

- Expense tracking and note management using JSON.
- Simple text-based menu interface.

Out of Scope:

- Graphical User Interface (GUI).
- Database integration.
- Cloud storage and deployment.

Bibliography

1. Python Official Documentation –
<https://docs.python.org/>
2. W3Schools Python Tutorial –
<https://www.w3schools.com/python/>
3. GeeksforGeeks –
<https://www.geeksforgeeks.org/python-programming-language/>
4. Real Python – <https://realpython.com/>