

# CAPSTONE PROJECT

## PERSONAL NOTES MANAGER

**PRESENTED BY**

**STUDENT NAME:** RISHIT GHOSH

**COLLEGE NAME:** Geethanjali College of  
Engineering and Technology

**DEPARTMENT:** CSE – AI&ML

**EMAIL ID:** rishitghosh06@gmail.com



# OUTLINE:

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

# PROBLEM STATEMENT:

Students and professionals often struggle to organize personal notes effectively.

Traditional paper-based or scattered digital notes lead to:

- Difficulty in searching specific information
- Lack of structured storage
- Risk of losing important notes

There is a need for a lightweight, reliable, and persistent system to manage notes efficiently.

# PROPOSED SOLUTION:

## Overview

A lightweight, command-line tool to manage personal notes with simplicity and reliability.

## Key Features

- **Add Notes** – Quickly capture ideas or tasks
- **View Notes** – Display stored notes in the terminal
- **Search Notes** – Find notes by keywords instantly
- **Delete Notes** – Remove outdated or irrelevant entries

## Technical Highlights

- **Persistent Storage** – Notes saved in JSON for durability
- **Modular Design** – Helper functions ensure clean, maintainable code
- **Unit Tests** – Guarantee correctness and reliability

## Benefits

- Easy to manage
- Fast and searchable
- Safely stored for long-term use

# SYSTEM APPROACH:

## System Requirements

- Python **3.10+**
- Pure Python (no external dependencies)

## Libraries Used

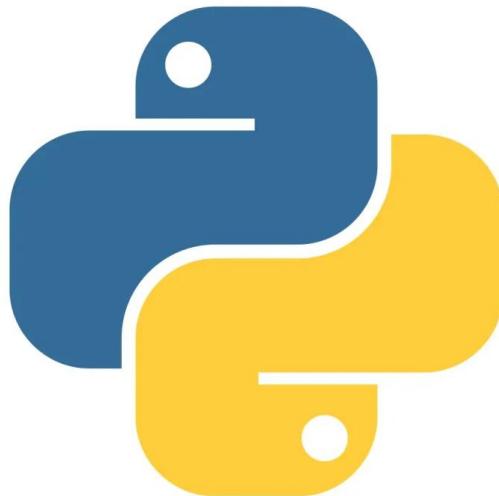
- **json** – Persistent storage of notes
- **os** – File handling and path operations
- **unittest** – Automated testing framework

## Folder Structure

- **main-scripts/** – Core application logic
- **utils/** – Helper functions for file operations
- **tests/** – Unit tests for reliability
- **exports/** – Screenshots, PDFs, and outputs

## Highlights

- Lightweight & dependency-free
- Clean modular organization
- Easy to test, maintain, and extend



# ALGORITHM & DEPLOYMENT:

## Algorithm Flow

1. **Initialize Storage** – Create notes\_storage.json if missing
2. **Add Note** – Append *title* + *content* to JSON
3. **View Notes** – Display all stored notes
4. **Search Notes** – Match keywords in title/content
5. **Delete Note** – Remove note by title

## Deployment

- **Run Application** -> “python -m main-scripts.notes\_manager”
- **Run Tests** -> “python -m unittest tests.test\_notes”

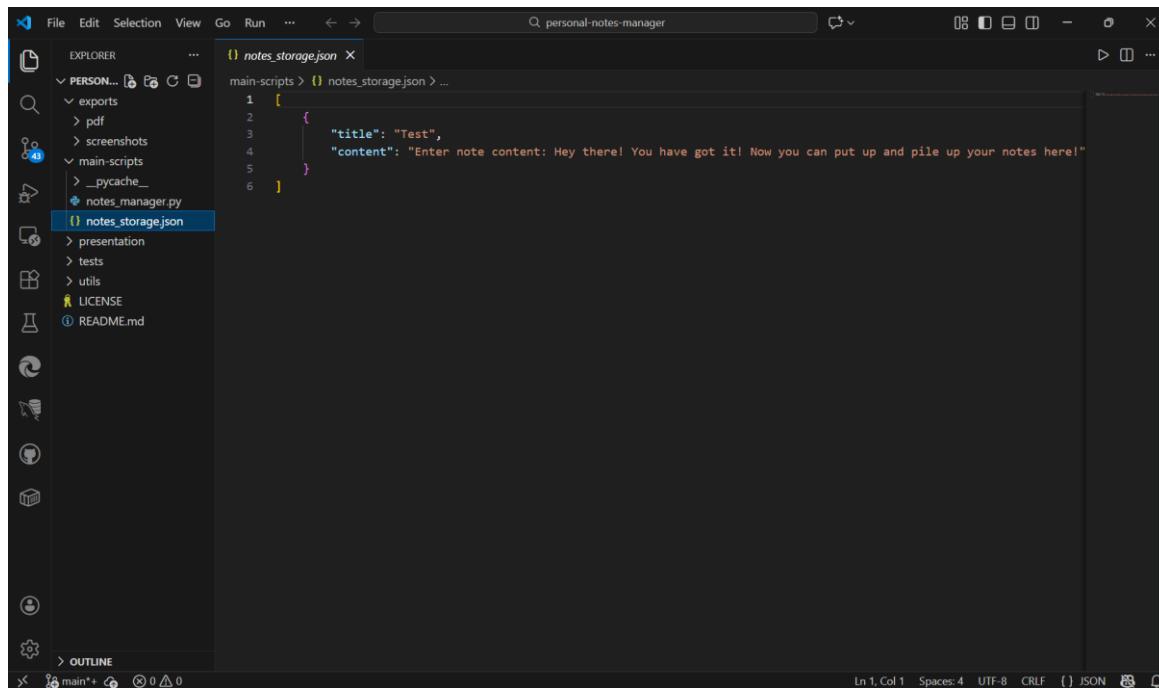
## Highlights

- Simple, intuitive workflow
- CLI-friendly execution
- Reliable testing ensures stability

# RESULT:

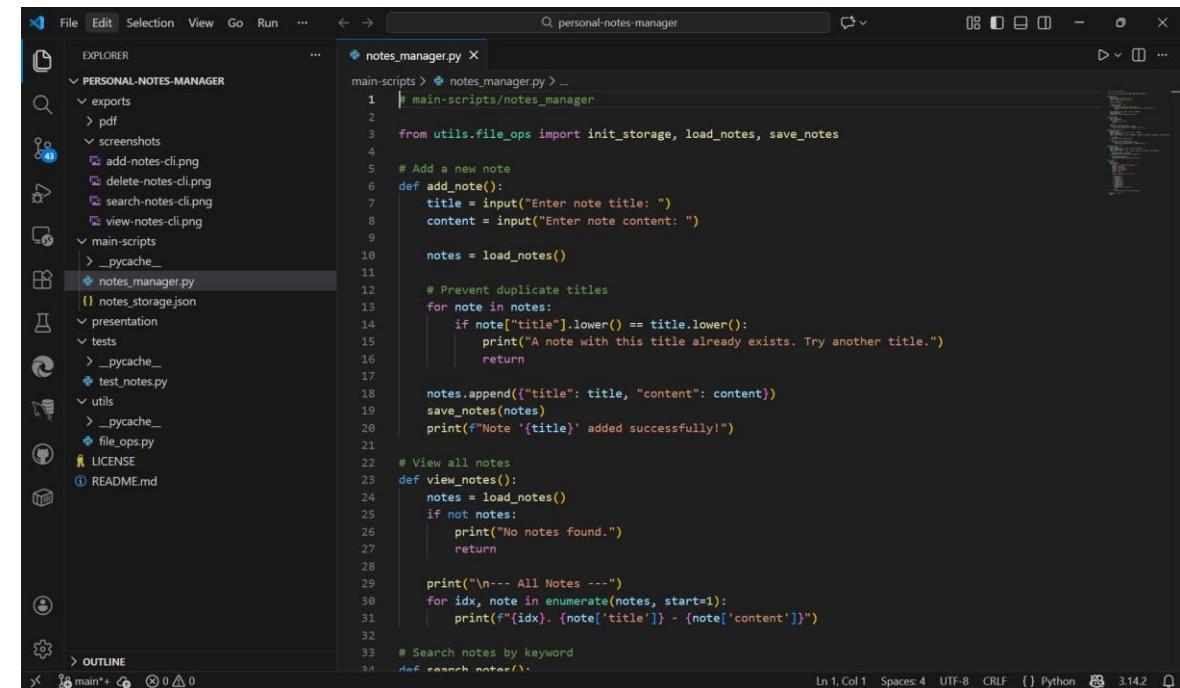
The Personal Notes Manager successfully demonstrated all core features – Add, View, Search, and Delete – with persistent JSON storage.

Unit tests executed with Python's unittest framework confirmed reliable functionality and accurate data handling.



```
notes_storage.json
1 [
2   {
3     "title": "Test",
4     "content": "Enter note content: Hey there! You have got it! Now you can put up and pile up your notes here!"
5   }
6 ]
```

.json file where the notes are saved.



```
notes_manager.py
1 # main-scripts/notes_manager
2
3 from utils.file_ops import init_storage, load_notes, save_notes
4
5 # Add a new note
6 def add_note():
7     title = input("Enter note title: ")
8     content = input("Enter note content: ")
9
10    notes = load_notes()
11
12    # Prevent duplicate titles
13    for note in notes:
14        if note["title"].lower() == title.lower():
15            print("A note with this title already exists. Try another title.")
16            return
17
18    notes.append({"title": title, "content": content})
19    save_notes(notes)
20    print(f"Note '{title}' added successfully!")
21
22 # View all notes
23 def view_notes():
24    notes = load_notes()
25    if not notes:
26        print("No notes found.")
27        return
28
29    print("\n--- All Notes ---")
30    for idx, note in enumerate(notes, start=1):
31        print(f"[{idx}]. {note['title']} - {note['content']}")
32
33 # Search notes by keyword
34 def search_notes():
35    pass
```

Code Sample.

# CONCLUSION:

## Problem Solved

- Tackles the issue of **scattered note-taking** by providing a centralized, structured solution

## Key Strengths

- **Lightweight** – Minimal setup, runs smoothly in any environment
- **Reliable** – JSON persistence + unit tests ensure stability
- **Easy to Use** – Simple CLI interface for quick note management
- **Fully Tested** – Verified functionality for confidence in daily use

## Outcome

- Demonstrates **practical application of Python** for workflow automation
- Enhances productivity through organized, searchable, and secure notes

# FUTURE SCOPE:

## Planned Features

- **Timestamps on Notes** – Automatically record creation and update times for better tracking
- **Export to PDF** – Generate shareable, professional note documents
- **GUI Interface** – User-friendly front end via **Tkinter** (desktop) or **Flask** (web)
- **Cloud Integration** – Sync and access notes anywhere using **Azure Functions**
- **Copilot Integration** – Smart suggestions to improve note organization and productivity

## Impact

- Enhances usability with multiple interfaces
- Enables portability and sharing of notes
- Provides intelligent assistance for smarter workflows
- Scales from personal use to cloud-connected productivity

# REFERENCES:

## Python Documentation

- json (docs.python.org in Bing) – JSON encoder/decoder
- os (docs.python.org in Bing) – Miscellaneous operating system interfaces
- unittest (docs.python.org in Bing) – Unit testing framework
- GitHub Repository – [Click here \[https://github.com/rajghosh06-dev/personal-notes-manager\]](https://github.com/rajghosh06-dev/personal-notes-manager)

# Thank You