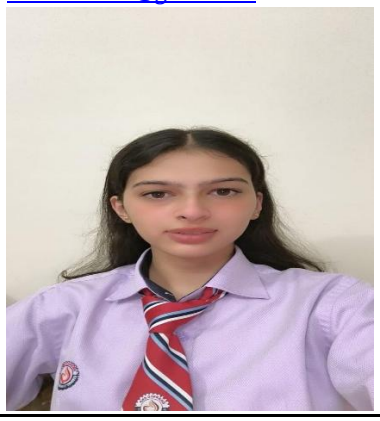
PROJECT AND TEAM INFORMATION

Project Title

Automatic File Organizer.

Student / Team Information

Team Name: *Team #*	*Bit by Bit* *PY-III-T067*
Team member 1 (Team Lead) *(Last Name, name: student ID: email, picture):*	*Nautiyal, Avani – 240121895* *Ava74knoty@gmail.com* 
Team member 2 *(Last Name, name: student ID: email, picture):*	*Chamoli,Pranjal – 24021077* *24021077@geu.ac.in* 

Team member 3 *(Last Name, name: student ID: email, picture):*	*Gupta,Avantika – 240222106* *240222106@geu.ac.in* 
Team member 4 *(Last Name, name: student ID: email, picture):*	*Singh,Saumya – 240222066* *240222066@geu.ac.in* 

PROPOSAL DESCRIPTION (10 pts)

Motivation (1 pt)

Managing large numbers of files on personal computers is often time-consuming and inefficient. Users struggle with cluttered folders, duplicate files, and difficulties in locating documents. This project aims to simplify file management by automatically categorizing, organizing, and providing advanced tools such as duplicate detection, preview, and search. A smart file organizer enhances productivity, saves time, and ensures better digital hygiene for both students and professionals.

State of the Art / Current solution (1 pt)

Currently, users manually organize files or rely on OS-level file explorers with limited sorting features. Some third-party tools exist, but they are either paid, platform-specific, or lack user-friendly customization such as custom rules, preview mode, or integrated statistics. Our project proposes a free, extensible, and user-centric alternative.

Project Goals and Milestones (2 pts)

Goals:

- Develop a cross-platform desktop tool for automatic file management.
- Provide customization, analytics, and user-friendly features beyond simple sorting.

Milestones:

1. **Core Modules:** Directory Selection, File Organization, Undo Operation.
2. **Extended Modules:**
    - Preview Mode (before moving files)
    - Custom Category Creation (user-defined rules)
    - Statistics Dashboard (file type analytics with charts)
    - Duplicate File Finder (hash-based)
    - Recycle Bin Integration (safe deletions)
    - Dark/Light Theme (UI flexibility)
    - File Search & Open (quick access)
3. Final integration, testing, and documentation.

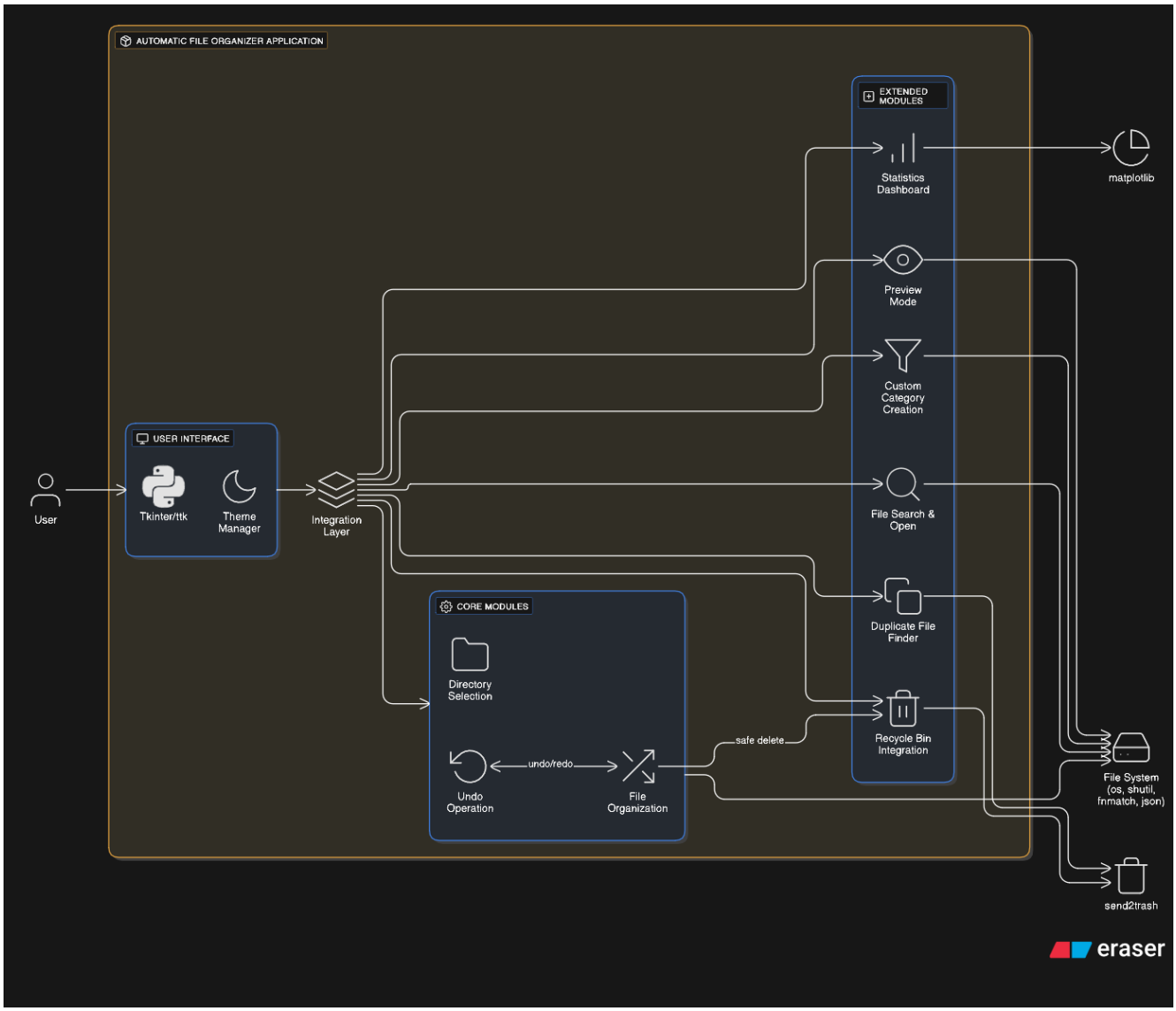
Project Approach (3 pts)

We will build the application in **Python** using:

- **Tkinter/ttk** for GUI and theme management.
- **os, shutil, fnmatch, json** for file operations and custom rules.
- **hashlib** for duplicate detection.
- **send2trash** for recycle bin integration.
- **matplotlib** for statistics visualization.

The project will follow a modular design. Each feature will be developed independently (assigned to team members) and later integrated. Testing will focus on responsiveness, error handling, and cross-platform compatibility

System Architecture (High Level Diagram)(2 pts)



Project Outcome / Deliverables (1 pts)

- A functional **Automatic File Organizer desktop application**.

- Features: automatic file sorting, undo, preview, duplicates detection, recycle bin safety, search & open, statistics dashboard, and customizable themes.

- Project report and presentation with demo

Applications

- Students: organize notes, assignments, PDFs.
- Professionals: manage office docs, project files, logs.
- Personal use: sort photos, music, downloads.
- Advanced: backups, duplicate removal, cloud sync

Assumptions

- Users will run the application on Windows/Linux with Python installed.
- Users will have permission to move/delete files in selected directories.
- Large folders may take extra time, so responsiveness will be handled with background threading.

References

- Python official documentation: https://docs.python.org
- Tkinter reference: https://docs.python.org/3/library/tkinter.html
- send2trash library: https://pypi.org/project/Send2Trash/
- Matplotlib: https://matplotlib.org
- Tutorials and resources on file handling and GUI programming in Python

Project Summary & Mentor Approval

Section	Summary
Project Title	Automatic File Organizer
Motivation	To simplify file management by automatically categorizing and organizing files, detecting duplicates, enabling previews, and providing advanced tools for productivity and digital hygiene.
State of the Art	Current solutions are either manual, limited to OS-level explorers, or third-party tools that lack free, customizable, and user-centric features. Our project offers a cross-platform, extensible alternative.
Project Goals & Milestones	Goals: **Build a desktop tool for automatic file management with customization, analytics, and user-friendly features.** Milestones: **Core modules (directory selection, organization, undo) → extended modules (preview, custom rules, stats dashboard, duplicate finder, recycle bin, search, themes) → final integration & testing.**
Project Approach	Developed in Python with Tkinter for GUI, `os`, `shutil`, `fnmatch`, `json` for file operations, `hashlib` for duplicate detection, `send2trash` for recycle bin, and `matplotlib` for statistics. Modular development ensures smooth integration and cross-platform compatibility.
System Architecture	User Input → File Scanner & Categorizer → Core Features (Sort, Undo, Search) + Extended Features (Preview, Duplicates, Stats, Theme) → Organized Output.
Deliverables	A functional desktop app with automatic sorting, undo, preview, duplicate detection, recycle bin safety, search & open, analytics dashboard, customizable themes, plus project report and demo.
Applications	Students: organize assignments/notes. Professionals: manage office docs/logs. Personal: sort media & downloads. Advanced: backups, duplicate removal, cloud sync.
Assumptions	Runs on Windows/Linux with Python installed, valid permissions for file operations, and large folders handled via background threading.