PROJECT WRITE-UP: TASK MANAGER

1. Problem Statement

Managing daily tasks is essential for productivity, yet individuals often struggle to track pending work in a structured manner. The objective of the Task Manager system is to enable users to securely store and manage their tasks. Each user should have private access to their own task list after authenticating with a username and password. The system must allow users to add, view, update, and delete tasks while maintaining persistent storage.

2. Approach

The solution was designed as a menu-driven, file-based Python application consisting of two core components:

a. **User Authentication:**

Users register with a unique username and password. Passwords are not stored in plain text; they are encrypted using SHA-256 hashing and written to a JSON file. During login, entered credentials are hashed and validated against stored values to ensure secure authentication.

b. **Task Management Module:**

Once logged in, users interact with a dedicated task file stored under their username. Tasks include a unique task ID, a description, and a status field (Pending or Completed). The system continuously reads and updates task data using file handling to ensure persistence even after program exit.

3. Summary of the Solution

The final Python implementation provides:

- User registration with secure password hashing
- Login validation using stored credentials
- Individual task storage per authenticated user
- Ability to add new tasks with auto-generated IDs
- View all tasks with status indicators
- Mark tasks as completed
- Delete unwanted tasks
- A loop-driven interactive menu for easy navigation
- Persistent storage using JSON files, eliminating the need for a database

This modular structure ensures that each user's data is private, secure, and retrievable. The application demonstrates authentication, file handling, and basic CRUD operations within a Python console environment.